LAMPIRAN

Lampiran 1. Persetujuan Etik

F-TI01

Formulir Permohonan Surat Pengantar Permintaan Data

Saya, yang bertanda tangan di bawah ini:

Nama	:	Raka Putra Atmaja
NIM	:	2215101010
Semester	:	6
Program Studi	:	Ilmu Komputer
No.Telp/HP	:	089668780490
Kegiatan	:	Skripsi
Dosen pembimbing	:	Kadek Yota Ernanda Aryanto, S.Kom., M.T., Ph.D.
Nama lomba/kegiatan	:	Skripsi
Judul skripsi/giat/lomba	:	SIMULASI IMPLEMENTASI GALERA CLUSTER UNTUK REPLIKASI SINKRONISASI PADA SISTEM BASIS DATA UNDIKSHA

Mengajukan permohonan untuk dibuatkan Surat Pengantar Permintaan Data kepada instansi/lembaga/perusahaan

Instansi/lembaga	:	UPA TIK
Pejabat tujuan	:	Kadek Surya Mahedy, S.T., M.Pd.
Data yang diperlukan	:	Informasi Arsitektur UPA TIK Undiksha khususnya pada system basis data yang sudah menggunakan metode Clustering pada database beserta permasalahannya

Demikian permohonan ini saya ajukan, atas perhatian dan bantuannya saya ucapkan terima kasih.

Singaraja, 30 Juni 2025

Menyetujui ,
Dosen Pembimbing

Kadek Yota Ernanda Aryanto, S.Kom., M.T., Ph.D.

Hormat saya,

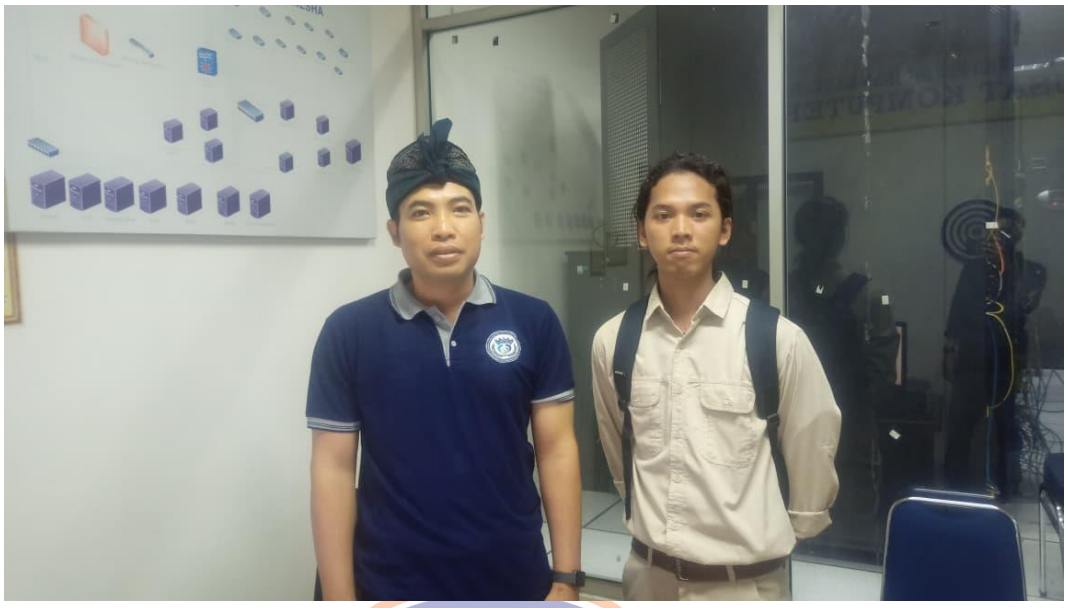
Raka Putra Atmaja

Formulir diserahkan ke tenaga kependidikan Jurusan Teknik Informatika.

Catatan :
- UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetakanya merupakan alat bukti hukum yang sah"
- Dokumen ini tertanda ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BsrE
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

Balai Sertifikasi Elektronik

Lampiran 2. Dokumentasi Wawancara Dengan Staff UPT TIK



Lampiran 3. Potongan Kode Uji Peforma Innodb

```
#!/bin/bash

DB_USER="root"
DB_PASS="node1"

ROUTER_HOST="192.168.56.10"
ROUTER_PORT="6446"

NODES=("192.168.56.10" "192.168.56.11"
"192.168.56.12")

DATA=10000
THREADS=20
TIME=30

# Cek MySQL Router
ROUTER_CHECK=$(mysql \
-u$DB_USER \
-p$DB_PASS \
-h$ROUTER_HOST \
-P$ROUTER_PORT \
-N \
-e "SELECT @@hostname,@@super_read_only" 2>/dev/null)

if [ -z "$ROUTER_CHECK" ]; then
    exit 1
fi
```

```
ROUTER_READONLY=$(echo "$ROUTER_CHECK" | awk '{print
$2}')

if [ "$ROUTER_READONLY" != "0" ]; then
    exit 1
fi

# Cek status node
for IP in "${NODES[@]}"; do

    STATUS=$(mysql \
    -u$DB_USER \
    -p$DB_PASS \
    -h$IP \
    -P3306 \
    -N \
    -e "SELECT @@super_read_only" 2>/dev/null)

done

# Cleanup database
sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
cleanup >/dev/null 2>&1

# Pengujian waktu sinkronisasi
START=$(date +%s%3N)

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
--tables=1 \
--table-size=$DATA \
prepare >/dev/null 2>&1 &

END2=0
END3=0

while true; do
```

```
    if [ "$END2" -eq 0 ]; then

        COUNT2=$(mysql \
        -u$DB_USER \
        -p$DB_PASS \
        -h192.168.56.11 \
        -P3306 \
        -N \
        -e  "SELECT  COUNT(*)  FROM  sbtest.sbtest1"
2>/dev/null)

        if [ "$COUNT2" = "$DATA" ]; then
            END2=$(date +%s%3N)
        fi
    fi

    if [ "$END3" -eq 0 ]; then
        COUNT3=$(mysql \
        -u$DB_USER \
        -p$DB_PASS \
        -h192.168.56.12 \
        -P3306 \
        -N \
        -e  "SELECT  COUNT(*)  FROM  sbtest.sbtest1"
2>/dev/null)

        if [ "$COUNT3" = "$DATA" ]; then
            END3=$(date +%s%3N)
        fi
    fi

    if [ "$END2" -ne 0 ] &&
       [ "$END3" -ne 0 ]; then
        break
    fi
done

TIME2=$((END2-START))
TIME3=$((END3-START))

if [ "$TIME2" -gt "$TIME3" ]; then
    SYNC_TIME=$TIME2
else
    SYNC_TIME=$TIME3
fi

# Pengujian performa

sysbench oltp_read_write \
```

```
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
--tables=1 \
--table-size=$DATA \
--threads=$THREADS \
--time=$TIME \
--delete_inserts=0 \
run > result.tmp

TPS=$(grep "transactions:" result.tmp |
awk '{print $3}' | tr -d '(')

LATENCY=$(grep "95th percentile:" result.tmp |
awk '{print $3}')

echo "Replication : $SYNC_TIME ms"
echo "TPS         : $TPS"
echo "Latency     : $LATENCY ms"
```

Lampiran 4. Potongan Kode Uji Peforma *Galera Cluster*

```
#!/bin/bash

DB_USER="testuser"
DB_PASS="test123"

NODES=("192.168.56.10"                    "192.168.56.11"
"192.168.56.12")
TARGET="192.168.56.10"

WRITE_PORT=3307
NODE_PORT=3306

DATA=700
THREAD=16
TIME=30

# Deteksi node aktif
ACTIVE=()

for IP in "${NODES[@]}"; do
    if mysqladmin -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT ping &>/dev/null; then

        STATE=$(mysql -u$DB_USER -p$DB_PASS \
```

```
        -h$IP -P$NODE_PORT -N -B \
        -e "SHOW GLOBAL STATUS LIKE
'wsrep_local_state_comment'" |
        awk '{print $2}')

        READY=$(mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT -N -B \
        -e "SHOW GLOBAL STATUS LIKE 'wsrep_ready'" |
        awk '{print $2}')

        if [[ "$STATE" == "Synced" &&
              "$READY" == "ON" ]]; then
            ACTIVE+=("$IP")
        fi
    fi
done

# Persiapan database
mysql -u$DB_USER -p$DB_PASS \
-h$TARGET -P$WRITE_PORT \
-e "CREATE DATABASE IF NOT EXISTS sbtest"

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$TARGET \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
cleanup

# Pengujian replikasi
START=$(date +%s%3N)

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$TARGET \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
--table-size=$DATA prepare

for IP in "${ACTIVE[@]}"; do
    while true; do
        COUNT=$(mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT -N -B \
        -e "SELECT COUNT(*) FROM sbtest.sbtest1")
```

```
        [[ "$COUNT" == "$DATA" ]] && break
        sleep 0.1
    done
done

END=$(date +%s%3N)
REPLICATION_TIME=$((END-START))

# Pengujian performa
sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$TARGET \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
--threads=$THREAD \
--time=$TIME run > result.tmp

TPS=$(grep transactions result.tmp | awk '{print $3}')
LATENCY=$(grep "95th percentile" result.tmp | awk
'{print $3}')

echo "Replication : $REPLICATION_TIME ms"
echo "TPS         : $TPS"
echo "Latency     : $LATENCY ms"
```

Lampiran 5. Potongan Kode *InnoDb 2 Node*

```
#!/bin/bash

# Konfigurasi
DB_USER="root"
DB_PASS="node1"

ROUTER_HOST="192.168.56.10"
ROUTER_PORT="6446"

NODES=("192.168.56.10""192.168.56.11"
"192.168.56.12")

DATA=7500
THREAD=20
TIME=30

# Deteksi node aktif dan primary
ACTIVE_NODES=()
PRIMARY_NODE=""
```

```
for IP in "${NODES[@]}"; do

    if mysqladmin -u$DB_USER -p$DB_PASS \
        -h$IP -P3306 ping &>/dev/null; then

        READ_ONLY=$(mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P3306 -N \
        -e "SELECT @@super_read_only")
        ACTIVE_NODES+=("$IP")
        if [ "$READ_ONLY" = "0" ]; then
            PRIMARY_NODE="$IP"
        fi
    fi
done

if [ ${#ACTIVE_NODES[@]} -lt 2 ] ||
   [ -z "$PRIMARY_NODE" ]; then
    exit 1
fi

# Verifikasi MySQL Router
ROUTER_STATUS=$(mysql \
-u$DB_USER \
-p$DB_PASS \
-h$ROUTER_HOST \
-P$ROUTER_PORT \
-N \
-e "SELECT @@super_read_only")

if [ "$ROUTER_STATUS" != "0" ]; then
    exit 1
fi

# Membersihkan data sebelumnya
sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
cleanup

# Pengujian waktu sinkronisasi
START=$(date +%s%3N)

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
```

```
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
--tables=1 \
--table-size=$DATA \
prepare

for IP in "${ACTIVE_NODES[@]}"; do

    while true; do
        COUNT=$(mysql \
        -u$DB_USER \
        -p$DB_PASS \
        -h$IP \
        -P3306 \
        -N \
        -e "SELECT COUNT(*) FROM sbtest.sbtest1")
        [ "$COUNT" = "$DATA" ] && break
        sleep 0.1
    done
done

END=$(date +%s%3N)
REPLICATION_TIME=$((END-START))

# Pengujian performa
sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$ROUTER_HOST \
--mysql-port=$ROUTER_PORT \
--mysql-db=sbtest \
--threads=$THREAD \
--time=$TIME \
run > result.tmp

TPS=$(grep "transactions:" result.tmp | awk '{print
$3}')
LATENCY=$(grep "95th percentile:" result.tmp | awk
'{print $3}')

echo "Primary       : $PRIMARY_NODE"
echo "Active Nodes  : ${ACTIVE_NODES[*]}"
echo "Replication   : $REPLICATION_TIME ms"
echo "TPS           : $TPS"
echo "Latency       : $LATENCY ms"
```

Lampiran 6. Potongan Kode *Galera Cluster* 2 *Node*

```
#!/bin/bash

DB_USER="testuser"
DB_PASS="test123"

ALL_NODES=("192.168.56.10"              "192.168.56.11"
"192.168.56.12")

TARGET_WRITE="192.168.56.10"
WRITE_PORT=3307
NODE_PORT=3306

DATA=7500
THREADS=20
TIME=30

ACTIVE_NODES=()

for IP in "${ALL_NODES[@]}"; do
    if mysqladmin -u$DB_USER -p$DB_PASS \
    -h$IP -P$NODE_PORT ping &>/dev/null; then

        STATE=$(mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT -N -B \
        -e      "SHOW      GLOBAL      STATUS      LIKE
'wsrep_local_state_comment'" |
        awk '{print $2}')

        READY=$(mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT -N -B \
        -e "SHOW GLOBAL STATUS LIKE 'wsrep_ready'" |
        awk '{print $2}')

        if [ "$STATE" = "Synced" ] &&
           [ "$READY" = "ON" ]; then
            ACTIVE_NODES+=("$IP")
        fi
    fi
done

if [ ${#ACTIVE_NODES[@]} -lt 2 ]; then
    exit 1
fi

mysql -u$DB_USER -p$DB_PASS \
-h$TARGET_WRITE -P$WRITE_PORT \
-e "CREATE DATABASE IF NOT EXISTS sbtest"
```

```
sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$TARGET_WRITE \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
cleanup

START=$(date +%s%3N)

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
--mysql-password=$DB_PASS \
--mysql-host=$TARGET_WRITE \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
--tables=1 \
--table-size=$DATA \
prepare >/dev/null 2>&1 &

COMPLETED_NODES=()
LAST_END=0

while [ ${#COMPLETED_NODES[@]} -lt ${#ACTIVE_NODES[@]}
]; do
    for IP in "${ACTIVE_NODES[@]}"; do
        if [[ ! " ${COMPLETED_NODES[@]} " =~ " ${IP} "
]]; then
            COUNT=$(mysql -u$DB_USER -p$DB_PASS \
            -h$IP -P$NODE_PORT -N -B \
            -e "SELECT COUNT(*) FROM sbtest.sbtest1")
            if [ "$COUNT" = "$DATA" ]; then
                END=$(date +%s%3N)
                COMPLETED_NODES+=("$IP")
                LAST_END=$END
            fi
        fi
    done
    sleep 0.1
done

SYNC_TIME=$((LAST_END-START))

sysbench oltp_read_write \
--db-driver=mysql \
--mysql-user=$DB_USER \
```

```
--mysql-password=$DB_PASS \
--mysql-host=$TARGET_WRITE \
--mysql-port=$WRITE_PORT \
--mysql-db=sbtest \
--tables=1 \
--table-size=$DATA \
--threads=$THREADS \
--time=$TIME \
--delete_inserts=0 \
--db-ps-mode=disable \
run > output_sysbench.tmp

TPS=$(grep "transactions:" output_sysbench.tmp |
awk '{print $3}' | tr -d '(')

LATENCY=$(grep "95th percentile:" output_sysbench.tmp
|
awk '{print $3}')

echo "Target : $TARGET_WRITE:$WRITE_PORT"
echo "Active Nodes : ${ACTIVE_NODES[*]}"
echo "Sync Time : $SYNC_TIME ms"
echo "TPS : $TPS"
echo "Latency : $LATENCY ms"
```

Lampiran 7. Potongan Kode *Failover InnoDb*

```
#!/bin/bash

DB_USER="root"
DB_PASS="node1"

NODES=("192.168.56.10" "192.168.56.11"
"192.168.56.12")

ROUTER_HOST="192.168.56.10"
ROUTER_PORT="6446"

format_ts() {
    echo "$1" | sed -E 's/.*([0-9]{2}\.[0-9]{3})/\1/'
}

# Mencari primary aktif
OLD_PRIMARY=""

for IP in "${NODES[@]}"; do

    STATUS=$(mysql \
    -u$DB_USER \
```

```
    -p$DB_PASS \
    -h$IP \
    --connect-timeout=1 \
    -sN \
    -e "SELECT @@super_read_only" 2>/dev/null)

    if [ "$STATUS" = "0" ]; then
        OLD_PRIMARY=$IP
        break
    fi
done

if [ -z "$OLD_PRIMARY" ]; then
    exit 1
fi
# Menunggu primary gagal
while true; do

    if ! mysql \
    -u$DB_USER \
    -p$DB_PASS \
    -h$OLD_PRIMARY \
    --connect-timeout=1 \
    -e "SELECT 1" &>/dev/null; then
        T_DOWN=$(date +%s.%3N)
        break
    fi

    sleep 0.05
done

# Deteksi primary baru
NEW_PRIMARY=""

while [ -z "$NEW_PRIMARY" ]; do

    for IP in "${NODES[@]}"; do

        [ "$IP" = "$OLD_PRIMARY" ] && continue

        STATUS=$(mysql \
        -u$DB_USER \
        -p$DB_PASS \
        -h$IP \
        --connect-timeout=1 \
        -sN \
        -e "SELECT @@super_read_only" 2>/dev/null)
```

```
        if [ "$STATUS" = "0" ]; then

            ROUTER=$(mysql \
            -u$DB_USER \
            -p$DB_PASS \
            -h$ROUTER_HOST \
            -P$ROUTER_PORT \
            --connect-timeout=1 \
            -sN \
            -e "SELECT @@super_read_only" 2>/dev/null)

            if [ "$ROUTER" = "0" ]; then
                T_UP=$(date +%s.%3N)
                NEW_PRIMARY=$IP
                break
            fi
        fi
    done
    sleep 0.05
done

RTO=$(echo "($T_UP-$T_DOWN)*1000" | bc)

# Menunggu node lama kembali
while true; do

    if mysql \
    -u$DB_USER \
    -p$DB_PASS \
    -h$OLD_PRIMARY \
    --connect-timeout=1 \
    -e "SELECT 1" &>/dev/null; then

        T_REC_START=$(date +%s.%3N)
        break
    fi
    sleep 0.1
done

# Menunggu node siap kembali
while true; do

    STATUS=$(mysql \
    -u$DB_USER \
    -p$DB_PASS \
    -h$OLD_PRIMARY \
    --connect-timeout=1 \
    -sN \
    -e "SELECT @@read_only" 2>/dev/null)
```

```
    if [ "$STATUS" = "1" ] ||
       [ "$STATUS" = "0" ]; then

        T_REC_END=$(date +%s.%3N)
        break
    fi
    sleep 0.1
done

MTTR=$(echo "($T_REC_END-$T_REC_START)*1000" | bc)

echo "Primary Lama : $OLD_PRIMARY"
echo "Primary Baru : $NEW_PRIMARY"
echo "RTO          : $RTO ms"
echo "MTTR         : $MTTR ms"
```

Lampiran 8. Potongan Kode *Failover Galera*

```
#!/bin/bash

DB_USER="testuser"
DB_PASS="test123"

NODES=("192.168.56.10"          "192.168.56.11"
"192.168.56.12")
NODE_PORT=3306

HAPROXY_HOST="192.168.56.10"
HAPROXY_PORT=3307

format_ts() {
    echo "$1" | sed -E 's/.*([0-9]{2}\.[0-9]{3})/\1/'
}

cek_node_synced() {

    IP=$1
    READY=$(mysql -u$DB_USER -p$DB_PASS \
    -h$IP -P$NODE_PORT \
    --connect-timeout=1 -sN \
    -e "SHOW GLOBAL STATUS LIKE 'wsrep_ready'" |
    awk '{print $2}')

    STATE=$(mysql -u$DB_USER -p$DB_PASS \
    -h$IP -P$NODE_PORT \
    --connect-timeout=1 -sN \
    -e      "SHOW      GLOBAL      STATUS      LIKE
'wsrep_local_state_comment'" |
```

```
    awk '{print $2}')
    CLUSTER=$(mysql -u$DB_USER -p$DB_PASS \
    -h$IP -P$NODE_PORT \
    --connect-timeout=1 -sN \
    -e       "SHOW       GLOBAL       STATUS       LIKE
'wsrep_cluster_status'" |
    awk '{print $2}')
    if [ "$READY" = "ON" ] &&
       [ "$STATE" = "Synced" ] &&
       [ "$CLUSTER" = "Primary" ]; then
        return 0
    fi
    return 1
}
cek_haproxy_write() {
mysql -u$DB_USER -p$DB_PASS \
-h$HAPROXY_HOST \
-P$HAPROXY_PORT <<EOF &>/dev/null

CREATE DATABASE IF NOT EXISTS ha_test;

CREATE TABLE IF NOT EXISTS ha_test.ha_check(
id INT AUTO_INCREMENT PRIMARY KEY,
waktu TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO ha_test.ha_check() VALUES();
EOF
}
ACTIVE_NODES=()
for IP in "${NODES[@]}"; do
    if cek_node_synced "$IP"; then
        ACTIVE_NODES+=("$IP")
    fi
done

if [ ${#ACTIVE_NODES[@]} -lt 2 ]; then
    exit 1
fi
if ! cek_haproxy_write; then
    exit 1
fi
# Pengujian Failover RTO
OLD_NODE=""
while [ -z "$OLD_NODE" ]; do
    for IP in "${ACTIVE_NODES[@]}"; do
        if ! mysql -u$DB_USER -p$DB_PASS \
        -h$IP -P$NODE_PORT \
        --connect-timeout=1 \
```

```
        -e "SELECT 1" &>/dev/null; then
            T_DOWN=$(date +%s.%3N)
            OLD_NODE=$IP
            break
        fi
    done
    sleep 0.05
done
while true; do
    if cek_haproxy_write; then
        T_UP=$(date +%s.%3N)
        break
    fi
    sleep 0.05
done

RTO_MS=$(echo "($T_UP-$T_DOWN)*1000" | bc)
while true; do

    if mysql -u$DB_USER -p$DB_PASS \
    -h$OLD_NODE -P$NODE_PORT \
    --connect-timeout=1 \
    -e "SELECT 1" &>/dev/null; then
        T_REC_START=$(date +%s.%3N)
        break
    fi
    sleep 0.1
done

while true; do
    if cek_node_synced "$OLD_NODE"; then
        T_REC_END=$(date +%s.%3N)
        break
    fi
    sleep 0.1
done

MTTR_MS=$(echo "($T_REC_END-$T_REC_START)*1000" | bc)

echo "Node Failed : $OLD_NODE"
echo "RTO         : $RTO_MS ms"
echo "MTTR        : $MTTR_MS ms"
```

Lampiran 9. Potongan Kode Analisis *Wilcoxon*

```
import math
import numpy as np
import pandas as pd
from scipy.stats import wilcoxon
```

```
kondisi = "2 Node Aktif"
skala_data = 7500

data = {
    "Throughput": {
        "InnoDB":
[14.09,16.08,13.38,16.16,12.19,14.56,13.98,13.40,14.7
0,14.25],
        "Galera":
[10.04,11.96,12.12,12.96,14.10,13.66,12.22,12.34,12.1
0,13.13],
        "lebih_baik": "lebih_besar"
    },
    "Latency": {
        "InnoDB":
[3841.98,2932.60,3267.19,3208.88,4599.99,2932.60,3095
.38,3639.94,3911.79,3208.88],
        "Galera":
[3706.08,2880.27,3208.88,3511.19,2449.36,2728.81,2985
.89,2828.87,2778.39,2632.28],
        "lebih_baik": "lebih_kecil"
    },
    "CPU": {
        "InnoDB":
[82.4,82.6,82.9,82.8,74.6,79.3,79.0,78.4,77.6,81.9],
        "Galera":
[77.5,75.2,81.4,78.2,79.8,76.3,80.6,75.7,81.7,77.2],
        "lebih_baik": "lebih_kecil"
    },
    "Memory": {
        "InnoDB":
[47.5,47.6,47.7,48.0,48.9,48.2,48.5,48.7,48.7,48.9],
        "Galera":
[28.6,28.7,29.1,29.2,29.6,30.7,30.6,30.6,31.2,31.2],
        "lebih_baik": "lebih_kecil"
    },
    "Waktu Replikasi": {
        "InnoDB":
[4313,4046,4497,4279,4278,4670,3505,4251,4743,4066],
        "Galera":
[6641,5811,6987,6916,5852,5878,5567,5297,6610,5714],
        "lebih_baik": "lebih_kecil"
    }
}

def kategori_effect_size(r):
    if r <= 0.30:
        return "Efek kecil"
```

```
    elif r <= 0.50:
        return "Efek sedang"
    return "Efek besar"

def tentukan_cluster(mean_i, mean_g, p, arah):
    if p >= 0.05:
        return "Tidak ditentukan"
    if arah == "lebih_besar":
        return  "Galera"  if  mean_g  >  mean_i  else
"InnoDB"
    return "Galera" if mean_g < mean_i else "InnoDB"

hasil = []
for metrik, nilai in data.items():
    innodb = np.array(nilai["InnoDB"], dtype=float)
    galera = np.array(nilai["Galera"], dtype=float)

    if len(innodb) != len(galera):
        raise ValueError(f"Jumlah data {metrik} tidak
sama.")
    if len(innodb) != 10:
        raise  ValueError(f"Jumlah  iterasi  {metrik}
harus 10.")

    uji = wilcoxon(
        innodb,
        galera,
        alternative="two-sided",
        zero_method="wilcox",
        method="approx"
    )

    mean_i = np.mean(innodb)
    mean_g = np.mean(galera)

    p = uji.pvalue
    effect       =       abs(uji.zstatistic)       /
math.sqrt(len(innodb))

    hasil.append({
        "Kondisi": kondisi,
        "Skala Data": skala_data,
        "Metrik": metrik,
        "Rata-rata InnoDB": round(mean_i,2),
        "Rata-rata Galera": round(mean_g,2),
        "p-value": round(p,4),
        "Effect Size": round(effect,2),
        "Kategori                                Efek":
kategori_effect_size(effect),
```

```
        "Keputusan": "Signifikan" if p < 0.05 else
"Tidak signifikan",
        "Cluster Unggul": tentukan_cluster(
            mean_i,
            mean_g,
            p,
            nilai["lebih_baik"]
        )
    })

hasil_df = pd.DataFrame(hasil)
print("\nHASIL UJI WILCOXON SIGNED-RANK TEST")
print("=" * 100)

print(hasil_df.to_string(index=False))
```