

LAMPIRAN

Lampiran 1 Surat permohonan data ke UPA TIK Undiksha



KEMENTERIAN PENDIDIKAN TINGGI,
SAINS, DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
Jalan Udayana Nomor 11 Singaraja Bali
Laman: <http://ftk.undiksha.ac.id>

Nomor : 516/UN48.11.1/KM/2025

Singaraja, 25 Pebruari 2025

Perihal : Surat Permohonan Pengambilan Data

Yth. Kepala UPA TIK Undiksha
di Tempat

Dengan hormat, sehubungan dengan proses penyelesaian Tugas Akhir/Skripsi, maka melalui surat ini kami mohon Bapak/Ibu berkenan memberikan data yang terkait dengan data yang dibutuhkan. Adapun mahasiswa yang akan melakukan pengambilan data seperti tersebut di bawah ini:

Nama : Ketut Saka Pradipta
NIM : 2115051008
Program Studi : Pendidikan Teknik Informatika
Jurusan : Teknik Informatika
Data yang dibutuhkan : data yang terkait dengan jumlah server yang dimiliki, terjadi gangguan pada server terkait frekuensi gangguan dan overload pengguna, riwayat downtime server, implementasi kontainerisasi pada server, teknologi ataupun system monitoring yang digunakan pada server
Judul Penelitian : Implementasi Kubernetes Cluster untuk High Availability dan Load Balancing pada Sistem Informasi Akademik Universitas Pendidikan Ganesha

Demikian kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terima kasih.

An Dekan
Wakil Dekan Bidang Akademik,

Made Windu Antara Kesiman
NIP 198211112008121001



516

KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
JURUSAN TEKNIK INFORMATIKA

Jalan Udayana Singaraja-Bali Kode Pos 81116
Tlp. (0362) 22570 Fax. (0362) 25735
Laman: www.undiksha.ac.id

Nomor : 92/UN48.11.5/KM/2025
Perihal : Surat Permohonan Pengambilan Data
Lampiran : -

Singaraja, 25 Februari 2025

Yth. Dekan FTK
Universitas Pendidikan Ganesha
Di tempat

Dengan hormat,
Sehubungan dengan proses penyelesaian Tugas Akhir / Skripsi yang dilaksanakan oleh saudara mahasiswa:

Nama	: Ketut Saka Pradipta
Nim	: 2115051008
Prodi/Jurusan	: Pendidikan Teknik Informatika / Teknik Informatika
Instansi yg ditujui	: UPT TIK Undiksha
Jabatan yang dituju	: Kepala UPT TIK Undiksha
Data yang dibutuhkan	: Data yang terkait seperti jumlah server yang dimiliki, terjadi gangguan pada server terkait frekuensi gangguan dan overload pengguna, riwayat downtime server, implementasi kontainerisasi pada server, teknologi ataupun system monitoring yang digunakan pada server
Judul	: Implementasi Kubernetes cluster untuk High Availability dan Load balancing pada Sistem Informasi Akademik Universitas Pendidikan Ganesha

Bersama ini kami mohonkan kepada Bapak untuk berkenan memfasilitasi kebutuhan data untuk Tugas Akhir / Skripsi mahasiswa yang bersangkutan.
Demikian kami sampaikan. Atas perhatian dan kerjasama Bapak, kami ucapkan terima kasih.

Ketua Jurusan Teknik Informatika,



Putu Hendra Suputra
NIP. 198212222006041001



Balai
Sertifikasi
Elektronik

Catatan :

- UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1 "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya 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

Lampiran 2 Dokumentasi Foto Kegiatan





Lampiran 3 Siak Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: laravel-app
  namespace: siak-system
spec:
  replicas: 2
  revisionHistoryLimit: 2
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 0
  selector:
    matchLabels:
      app: laravel-app
  template:
    metadata:
      labels:
        app: laravel-app
    spec:
      terminationGracePeriodSeconds: 3
      # "HA + scalable"
      topologySpreadConstraints:
        - maxSkew: 2
          topologyKey: "kubernetes.io/hostname"
          whenUnsatisfiable: "ScheduleAnyway"
          labelSelector:
            matchLabels:
              app: laravel-app
      affinity:
```

```
nodeAffinity:
  requiredDuringSchedulingIgnoredDuringExecution:
    nodeSelectorTerms:
      - matchExpressions:
          - key: "role"
            operator: "In"
            values:
              - worker
tolerations:
  - key: "node.kubernetes.io/not-ready"
    operator: "Exists"
    effect: "NoExecute"
    tolerationSeconds: 3
  - key: "node.kubernetes.io/unreachable"
    operator: "Exists"
    effect: "NoExecute"
    tolerationSeconds: 3
containers:
  - name: laravel-app
    image: docker.io/saka8631/laravel-siak:latest
    ports:
      - containerPort: 9000
    envFrom:
      - configMapRef:
          name: laravel-env
```


Lampiran 4 Siak Service

```
apiVersion: v1
kind: Service
metadata:
  name: laravel-app
  namespace: siak-system
spec:
  selector:
    app: laravel-app
  ports:
    - protocol: TCP
      port: 9000
      targetPort: 9000
```



Lampiran 5 Konfigurasi Kubernetes Cluster

Langkah 1: Persiapan Sistem

Update System

```
sudo apt update && sudo apt upgrade -y
```

Disable Swap

```
sudo swapoff -a  
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

Load Kernel Modules

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
overlay  
br_netfilter  
EOF  
sudo modprobe overlay  
sudo modprobe br_netfilter
```

Configure Sysctl

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf  
net.bridge.bridge-nf-call-iptables = 1  
net.bridge.bridge-nf-call-ip6tables = 1  
net.ipv4.ip_forward = 1  
EOF
```

```
sudo sysctl --system
```

Langkah 2: Install CRI-O Container Runtime

Add CRI-O Repository

```
OS="xUbuntu_24.04"  
VERSION="1.28"  
echo "deb [signed-by=/usr/share/keyrings/libcontainers-  
archive-keyring.gpg]  
https://download.opensuse.org/repositories/devel:kubic:lib  
containers:/stable/$OS/ /" | sudo tee  
/etc/apt/sources.list.d/devel:kubic:libcontainers:stable.lis  
t
```

```
echo "deb [signed-by=/usr/share/keyrings/libcontainers-crio-
archive-keyring.gpg]
https://download.opensuse.org/repositories/devel:/kubic:/lib
containers:/stable:/cri-o:/$VERSION/$OS/ /" | sudo tee
/etc/apt/sources.list.d/devel:kubic:libcontainers:stable:cri
-o:$VERSION.list
```

Install CRI-O

```
sudo apt update

sudo apt install cri-o cri-o-runc cri-tools -y
```

Start CRI-O Service

```
sudo systemctl enable crio

sudo systemctl start crio

sudo systemctl status crio
```

Langkah 3: Install Kubernetes Components

Add Kubernetes Repository

```
sudo curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-
keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-
keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /'
| sudo tee /etc/apt/sources.list.d/kubernetes.list
```

Install Packages

```
sudo apt update

sudo apt install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl
```

Verify Installation

```
kubectl version --client

kubeadm version
```

Langkah 4: Initialize Master Node

Initialize Cluster

```
sudo kubeadm init --pod-network-cidr=192.168.0.0/16
```

Configure kubectl

```
mkdir -p $HOME/.kube
```



```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Install Calico Network Plugin

```
kubectl create -f  
https://raw.githubusercontent.com/projectcalico/calico/v3.26  
.1/manifests/tigera-operator.yaml  
  
kubectl create -f  
https://raw.githubusercontent.com/projectcalico/calico/v3.26  
.1/manifests/custom-resources.yaml
```

Verify Cluster

```
kubectl get nodes  
kubectl get pods -n kube-system
```

Langkah 5: Join Worker Nodes

Get Join Command

On master node:

```
kubeadm token create --print-join-command
```

Join Worker Nodes

On each worker node:

```
sudo kubeadm join [MASTER_IP]:6443 --token [TOKEN] --  
discovery-token-ca-cert-hash sha256:[HASH]
```

Langkah 6: Verifikasi Cluster

Check Node Status

```
kubectl get nodes -o wide
```

Deploy Test Application

```
kubectl create deployment nginx --image=nginx  
kubectl expose deployment nginx --port=80 --type=LoadBalancer  
kubectl get pods,svc
```

Clean Up

```
kubectl delete deployment nginx  
kubectl delete service nginx
```

Troubleshooting

Common Issues

```
CRI-O not starting: Check logs with journalctl -u crio  
Node NotReady: Verify network connectivity and firewall rules  
Pod stuck in Pending: Check resource availability and network  
plugin status
```

Useful Commands

Check cluster status

```
kubectl cluster-info  
kubectl get componentstatuses
```

Check node details

```
kubectl describe node [NODE_NAME]
```

Check pod logs

```
kubectl logs [POD_NAME] -n [NAMESPACE]
```



Lampiran 6 Konfigurasi Prometheus dan Grafana

Langkah 1 – Instalasi Node Exporter Di Setiap Node (Master & Worker)

```
Wget
https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz

tar xvfz node_exporter-1.7.0.linux-amd64.tar.gz

sudo cp node_exporter-1.7.0.linux-amd64/node_exporter
/usr/local/bin/

# Buat service

sudo tee /etc/systemd/system/node_exporter.service >
/dev/null <<EOF

[Unit]

Description=Node Exporter

After=network.target


[Service]

User=nobody

ExecStart=/usr/local/bin/node_exporter

Restart=on-failure


[Install]

WantedBy=default.target

EOF

# Jalankan

sudo systemctl daemon-reexec

sudo systemctl enable --now node_exporter
```

Cek:

```
curl http://localhost:9100/metrics(atau via IP dari server
monitoring nanti)
```

LANGKAH 2 – Instalasi Prometheus di Server Monitoring

```
sudo useradd --no-create-home --shell /bin/false prometheus

# Download & install

wget
https://github.com/prometheus/prometheus/releases/download/v
2.52.0/prometheus-2.52.0.linux-amd64.tar.gz

tar xvf prometheus-2.52.0.linux-amd64.tar.gz

cd prometheus-2.52.0.linux-amd64

sudo cp prometheus promtool /usr/local/bin/

sudo mkdir -p /etc/prometheus /var/lib/prometheus

sudo cp -r consoles console_libraries /etc/prometheus/
```

LANGKAH 3 – Konfigurasi Prometheus untuk Scrape Remote Nodes

```
nano /etc/prometheus/prometheus.yml
```

Edit file konfigurasi seperti berikut:

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'node-exporter'
    static_configs:
      - targets: ['192.168.1.1:9100', '192.168.1.2:9100',
'192.168.1.3:9100']

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['192.168.1.1:30180', '192.168.1.2:30180',
'192.168.1.3:30180']

  - job_name: 'kubernetes-apiserver'
```



```

    scheme: https

    tls_config:

        insecure_skip_verify: true

    authorization:

        credentials:
eyJhbGciOiJSUzI1NiIsImtpZCI6InB5Yz1ScGhtTXhfenBQYzJ0VGJzLUVs
SEJ6ZEtuQzNudHNXck9i>

    static_configs:

        - targets: ['192.168.1.1:6443']

```

LANGKAH 4 – Jalankan Prometheus Sebagai Service

```
nano /etc/systemd/system/prometheus.service
```

Edit file konfigurasi seperti berikut:

```

[Unit]

Description=Prometheus Monitoring
Wants=network-online.target
After=network-online.target


[Service]
User=prometheus
ExecStart=/usr/local/bin/prometheus \
    --config.file=/etc/prometheus/prometheus.yml \
    --storage.tsdb.path=/var/lib/prometheus \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries


[Install]
WantedBy=multi-user.target

```

Berikan kepemilikan untuk file yang sudah dibuat lalu restart daemin dan enable promotheus :

```

sudo chown -R prometheus:prometheus /etc/prometheus
/var/lib/prometheus

sudo systemctl daemon-reexec

```

```
sudo systemctl enable --now Prometheus
```

LANGKAH 5 – Instalasi Grafana

1. Buat folder keyring (jika belum ada)

```
sudo mkdir -p /etc/apt/keyrings
```

2. Download dan simpan GPG key Grafana

```
curl -fsSL https://packages.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
```

3. Tambahkan repository Grafana dengan signed-by

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://packages.grafana.com/oss/deb stable main" | sudo tee /etc/apt/sources.list.d/grafana.list
```

4. Update & install Grafana

```
sudo apt update  
sudo apt install grafana -y
```

5. Jalankan Grafana

```
sudo systemctl enable --now grafana-server
```

Setelah ini, kamu bisa akses Grafana di:

```
http://<IP-SERVER>:3000  
Login: admin / admin
```