

# Referensi Cepat systemd

Manajemen layanan, unit, timer, dan journalctl

## Manajemen Layanan

### Perintah Layanan Dasar

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx # reload config
systemctl status nginx
```

### Enable / Disable

```
systemctl enable nginx # start at boot
systemctl disable nginx # remove from boot
systemctl enable --now nginx # enable + start
systemctl is-enabled nginx
```

### Status Layanan

<b>active (running)</b>	Layanan berjalan normal
<b>active (exited)</b>	Berjalan sekali dan selesai dengan sukses
<b>inactive (dead)</b>	Layanan dihentikan
<b>failed</b>	Layanan crash atau keluar dengan error
<b>activating</b>	Layanan sedang dijalankan

## File Unit

### Lokasi File Unit

<b>/etc/systemd/system/</b>	Unit buatan admin (prioritas tertinggi)
<b>/run/systemd/system/</b>	Unit yang dibuat saat runtime
<b>/usr/lib/systemd/system/</b>	Unit yang diinstal paket
<b>~/.config/systemd/user/</b>	Unit tingkat pengguna

### Unit Layanan Dasar

```
[Unit]
Description=My Application
After=network.target
[Service]
ExecStart=/usr/bin/myapp --config /etc/myapp.conf
Restart=on-failure
User=appuser
[Install]
WantedBy=multi-user.target
```

### Terapkan Perubahan

```
systemctl daemon-reload # reload unit files
systemctl restart myapp # apply changes
```

## Timer

### Unit Timer

```
[Unit]
Description=Run backup daily
[Timer]
OnCalendar=*. * * * * 02:00:00
Persistent=true
[Install]
WantedBy=timers.target
```

### Sintaks OnCalendar

<b>* * * * * 02:00:00</b>	Setiap hari pukul 02:00
<b>Mon * * * * * 09:00:00</b>	Setiap Senin pukul 09:00
<b>* * - 01 00:00:00</b>	Hari pertama setiap bulan
<b>hourly / daily / weekly</b>	Jadwal singkat

### Manajemen Timer

```
systemctl list-timers --all
systemctl start backup.timer
systemctl enable backup.timer
systemd-analyze calendar "daily"
```

## Target

### Target Umum

<b>multi-user.target</b>	Boot normal, multi-pengguna, tanpa GUI
<b>graphical.target</b>	Desktop GUI lengkap
<b>rescue.target</b>	Mode rescue single-user
<b>emergency.target</b>	Shell minimal, hanya root
<b>network-online.target</b>	Jaringan sudah terkonfigurasi penuh
<b>timers.target</b>	Semua unit timer siap

### Perintah Target

```
systemctl get-default
systemctl set-default multi-user.target
systemctl isolate rescue.target
systemctl list-dependencies graphical.target
```

## Journalctl

### Melihat Log

```
journalctl -u nginx # logs for unit
journalctl -u nginx -f # follow (tail)
journalctl -u nginx --no-pager
journalctl -b # current boot only
```

### Filter Log

```
journalctl --since "2026-03-01"
journalctl --since "1 hour ago"
journalctl -p err # errors and above
journalctl _PID=1234
```

### Level Prioritas

<b>emerg (0)</b>	Sistem tidak dapat digunakan
<b>alert (1)</b>	Tindakan segera diperlukan
<b>crit (2)</b>	Kondisi kritis
<b>err (3)</b>	Kondisi error
<b>warning (4)</b>	Kondisi peringatan
<b>info (6)</b>	Informasional
<b>debug (7)</b>	Pesan level debug

### Pemeliharaan Log

```
journalctl --disk-usage
journalctl --vacuum-size=500M
journalctl --vacuum-time=30d
```

## Jaringan

### networkctl

```
networkctl list
networkctl status eth0
networkctl up eth0
networkctl down eth0
```

### systemd-resolve

```
resolvectl status
resolvectl query example.com
resolvectl flush-caches
resolvectl statistics
```

### Tunggu Jaringan

```
# In unit file [Unit] section:
After=network-online.target
Wants=network-online.target
```

## Mount

### Unit Mount

```
[Unit]
Description=Mount data volume
[Mount]
What=/dev/sdb1
Where=/mnt/data
Type=ext4
Options=defaults,noatime
[Install]
WantedBy=multi-user.target
```

### Unit Automount

```
[Unit]
Description=Automount data on access
[Automount]
Where=/mnt/data
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
```

### Konvensi Penamaan

**/mnt/data** File unit: **mnt-data.mount**  
**/var/lib/app** File unit: **var-lib-app.mount**

Path mount dengan `/' diganti `-', tanda hubung awal dihilangkan

## Environment

### Mengatur Variabel Environment

```
[Service]
Environment=APP_ENV=production
Environment=PORT=8080
EnvironmentFile=/etc/myapp/env
```

### Format File Environment

```
# /etc/myapp/env
APP_ENV=production
DATABASE_URL=postgres://localhost/db
SECRET_KEY=changeme
```

## Hardening Layanan

<b>ProtectSystem=strict</b>	Filesystem hanya-baca kecuali path yang diizinkan
<b>ProtectHome=true</b>	Sembunyikan /home, /root, /run/ user
<b>NoNewPrivileges=true</b>	Cegah eskalasi hak istimewa
<b>PrivateTmp=true</b>	/tmp terisolasi untuk layanan
<b>ReadWritePaths=/var/lib/myapp</b>	Izinkan penulisan ke path tertentu

## Dependensi

### Direktif Urutan dan Ketergantungan

<b>After=b.service</b>	Mulai setelah b (hanya urutan)
<b>Before=b.service</b>	Mulai sebelum b (hanya urutan)
<b>Requires=b.service</b>	Dependensi keras; gagal jika b gagal
<b>Wants=b.service</b>	Dependensi lunak; tidak gagal jika b gagal
<b>BindsTo=b.service</b>	Berhenti ketika b berhenti
<b>Conflicts=b.service</b>	Tidak bisa berjalan bersamaan dengan b

### Inspeksi Dependensi

```
systemctl list-dependencies nginx
systemctl list-dependencies --reverse nginx
systemd-analyze dot nginx.service | dot -Tsvg > deps.svg
```

# Referensi Cepat systemd

---

## Pola Umum

---

### Kebijakan Restart

<b>Restart=no</b>	Tidak pernah restart (default)
<b>Restart=on-failure</b>	Restart jika keluar dengan kode non-nol
<b>Restart=always</b>	Selalu restart (untuk daemon)
<b>RestartSec=5</b>	Tunggu 5 detik sebelum restart
<b>StartLimitBurst=3</b>	Maksimum restart dalam interval
<b>StartLimitIntervalSec=60</b>	Interval untuk penghitungan burst

### Override Tanpa Mengedit

```
systemctl edit nginx # creates drop-in
# /etc/systemd/system/nginx.service.d/override.conf
systemctl cat nginx # show effective config
systemctl revert nginx # remove overrides
```

### Analisis Sistem

```
systemd-analyze # boot time
systemd-analyze blame # per-unit time
systemd-analyze critical-chain
systemctl list-units --failed
```