

Panduan Deployment CCTV Dashboard

Dokumentasi lengkap cara deploy aplikasi CCTV Dashboard di Ubuntu Server yang benar-benar kosong (fresh install).

Daftar Isi

1. [Persyaratan Sistem](#)
 2. [Instalasi Dependencies](#)
 3. [Setup Database PostgreSQL](#)
 4. [Clone dan Setup Project](#)
 5. [Konfigurasi Environment](#)
 6. [Setup Database Schema](#)
 7. [Build dan Run Aplikasi](#)
 8. [Setup PM2 Process Manager](#)
 9. [Setup Nginx Reverse Proxy dengan HTTP/2](#)
 10. [Setup SSL dengan Let's Encrypt](#)
 11. [Setup MediaMTX \(Media Server\)](#)
 12. [Troubleshooting](#)
-

Persyaratan Sistem

Minimum Hardware

- **CPU:** 2 cores (4+ cores direkomendasikan untuk streaming multiple kamera)
- **RAM:** 4GB minimum (8GB+ direkomendasikan)
- **Storage:** 20GB minimum (lebih banyak untuk recordings)
- **Network:** Bandwidth yang cukup untuk RTSP streams

Software

- Ubuntu Server 20.04 LTS atau 22.04 LTS
- Node.js 18.x atau lebih baru
- PostgreSQL 14 atau lebih baru
- Nginx (untuk reverse proxy dengan HTTP/2)
- PM2 (untuk process management)
- MediaMTX (untuk konversi RTSP/RTMP ke HLS)

Catatan Penting tentang Layout Kamera

Dashboard ini mendukung layout **2x2** (4 kamera) dan **3x3** (9 kamera) per halaman. Layout 4x4 dihilangkan karena limitasi browser:

- Browser membatasi **6 koneksi HTTP/1.1 per domain**
- 16 stream sekaligus akan menyebabkan antrian koneksi

- Menggunakan **HTTP/2** dapat mengatasi limitasi ini, namun tetap membebani bandwidth dan memory

Rekomendasi:

- Gunakan **HTTP/2** (wajib untuk performa optimal)
- Gunakan **sub-stream** (resolusi rendah) untuk live view grid
- Pagination otomatis tersedia jika kamera > kapasitas layout

Instalasi Dependencies

1. Update System

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

2. Install Node.js 18.x

```
# Install curl jika belum ada
sudo apt install -y curl

# Setup NodeSource repository untuk Node.js 18.x
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -

# Install Node.js
sudo apt install -y nodejs

# Verifikasi instalasi
node --version # Harus menampilkan v18.x.x
npm --version  # Harus menampilkan versi npm
```

3. Install Yarn (Package Manager)

```
# Install Yarn secara global
sudo npm install -g yarn

# Verifikasi instalasi
yarn --version
```

4. Install Build Tools

```
sudo apt install -y build-essential git
```

Setup Database PostgreSQL

1. Install PostgreSQL

```
# Install PostgreSQL
sudo apt install -y postgresql postgresql-contrib

# Start dan enable PostgreSQL service
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

2. Buat Database dan User

```
# Login sebagai postgres user
sudo -u postgres psql
```

Di dalam PostgreSQL shell, jalankan:

```
-- Buat user baru (ganti 'your_password' dengan password yang aman)
CREATE USER cctv_user WITH PASSWORD 'your_password';

-- Buat database
CREATE DATABASE cctv_dashboard;

-- Berikan hak akses penuh ke user
GRANT ALL PRIVILEGES ON DATABASE cctv_dashboard TO cctv_user;

-- Untuk PostgreSQL 15+, tambahkan juga:
c cctv_dashboard
GRANT ALL ON SCHEMA public TO cctv_user;

-- Keluar dari psql
q
```

3. Konfigurasi PostgreSQL untuk Koneksi Lokal

Edit file `pg_hba.conf` :

```
sudo nano /etc/postgresql/14/main/pg_hba.conf
```

Pastikan ada baris berikut (atau tambahkan):

```
# IPv4 local connections:
host      all             all             127.0.0.1/32      md5
```

Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

Clone dan Setup Project

1. Buat Direktori Aplikasi

```
# Buat direktori untuk aplikasi
sudo mkdir -p /var/www/cctv-dashboard
sudo chown $USER:$USER /var/www/cctv-dashboard

# Masuk ke direktori
cd /var/www/cctv-dashboard
```

2. Clone atau Copy Project

Jika menggunakan Git:

```
git clone <your-repository-url> .
```

Atau copy dari local machine:

```
# Dari local machine, gunakan scp:
scp -r /path/to/cctv_dashboard/nextjs_space/* user@server:/var/www/cctv-dashboard/
```

3. Install Dependencies

```
cd /var/www/cctv-dashboard
yarn install
```

Konfigurasi Environment

1. Buat File .env

```
nano .env
```

Isi dengan konfigurasi berikut:

```
# Database Connection
# Format: postgresql://USER:PASSWORD@HOST:PORT/DATABASE
DATABASE_URL="postgresql://cctv_user:your_password@localhost:5432/cctv_dashboard"

# NextAuth Configuration
# Generate secret dengan: openssl rand -base64 32
NEXTAUTH_SECRET="your-random-secret-key-here"
NEXTAUTH_URL="http://your-domain.com" # Ganti dengan domain/IP server Anda

# Node Environment
NODE_ENV="production"
```

2. Generate NEXTAUTH_SECRET

```
# Generate random secret key  
openssl rand -base64 32
```

Copy output dan paste ke `.env` file.

Setup Database Schema

1. Generate Prisma Client

```
yarn prisma generate
```

2. Push Schema ke Database

```
yarn prisma db push
```

Catatan: Project ini menggunakan `prisma db push` untuk sinkronisasi schema, bukan migrations. Jangan gunakan `yarn prisma migrate deploy` karena tidak ada migration files di project ini.

3. Seed Data Awal (Opsional)

Untuk mengisi database dengan data demo:

```
yarn prisma db seed
```

Ini akan membuat:

- Admin user: `john@doe.com` / `johndoe123`
 - Regular user: `user@test.com` / `testuser123`
 - 4 camera groups
 - 16 demo cameras
-

Build dan Run Aplikasi

1. Build Aplikasi

```
yarn build
```

2. Test Run (Development)

```
# Untuk testing  
yarn start
```

Akses `http://your-server-ip:3000` untuk memastikan aplikasi berjalan.

Setup PM2 Process Manager

PM2 digunakan untuk menjalankan aplikasi di background dan auto-restart jika server reboot.

1. Install PM2

```
sudo npm install -g pm2
```

2. Buat File Ecosystem PM2

```
nano ecosystem.config.js
```

Isi dengan:

```
module.exports = {  
  apps: [{  
    name: 'cctv-dashboard',  
    script: 'yarn',  
    args: 'start',  
    cwd: '/var/www/cctv-dashboard',  
    instances: 1,  
    autorestart: true,  
    watch: false,  
    max_memory_restart: '1G',  
    env: {  
      NODE_ENV: 'production',  
      PORT: 3000  
    }  
  }]  
};
```

3. Start Aplikasi dengan PM2

```
pm2 start ecosystem.config.js
```

4. Setup PM2 Startup

```
# Generate startup script  
pm2 startup  
  
# Ikuti instruksi yang ditampilkan (copy dan jalankan perintah sudo)  
  
# Save PM2 process list  
pm2 save
```

5. PM2 Commands Berguna

```
pm2 status          # Lihat status aplikasi  
pm2 logs cctv-dashboard # Lihat logs  
pm2 restart cctv-dashboard # Restart aplikasi  
pm2 stop cctv-dashboard # Stop aplikasi  
pm2 monit           # Monitor real-time
```

Setup Nginx Reverse Proxy dengan HTTP/2

HTTP/2 **sangat direkomendasikan** untuk streaming multiple kamera karena menghilangkan limitasi 6 koneksi per domain yang ada di HTTP/1.1.

1. Install Nginx

```
sudo apt install -y nginx
```

2. Buat Konfigurasi Nginx (HTTP saja - sebelum SSL)

```
sudo nano /etc/nginx/sites-available/cctv-dashboard
```

Konfigurasi awal (HTTP):

```
server {
    listen 80;
    server_name your-domain.com; # Ganti dengan domain atau IP Anda

    # Logging
    access_log /var/log/nginx/cctv-dashboard.access.log;
    error_log /var/log/nginx/cctv-dashboard.error.log;

    # Security headers
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection "1; mode=block";

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;

        # Timeout settings untuk streaming
        proxy_connect_timeout 60s;
        proxy_send_timeout 60s;
        proxy_read_timeout 60s;
    }

    # Untuk file uploads (jika ada)
    client_max_body_size 100M;
}
```

3. Enable Site dan Test Config

```
# Enable site
sudo ln -s /etc/nginx/sites-available/cctv-dashboard /etc/nginx/sites-enabled/

# Hapus default site (opsional)
sudo rm /etc/nginx/sites-enabled/default

# Test konfigurasi
sudo nginx -t

# Restart Nginx
sudo systemctl restart nginx
sudo systemctl enable nginx
```

4. Konfigurasi HTTP/2 (Setelah SSL Terpasang)

Setelah SSL certificate terpasang (lihat bagian SSL di bawah), update konfigurasi untuk mengaktifkan HTTP/2:

```
sudo nano /etc/nginx/sites-available/cctv-dashboard
```

Konfigurasi lengkap dengan **HTTP/2**:


```

# Redirect HTTP ke HTTPS
server {
    listen 80;
    server_name your-domain.com;
    return 301 https://$server_name$request_uri;
}

# HTTPS dengan HTTP/2
server {
    listen 443 ssl http2; # HTTP/2 enabled!
    server_name your-domain.com;

    # SSL Certificate (akan diisi oleh Certbot)
    ssl_certificate /etc/letsencrypt/live/your-domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/your-domain.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    # Logging
    access_log /var/log/nginx/cctv-dashboard.access.log;
    error_log /var/log/nginx/cctv-dashboard.error.log;

    # Security headers
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection "1; mode=block";
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

    # Dashboard Application
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;

        # Timeout settings untuk streaming
        proxy_connect_timeout 60s;
        proxy_send_timeout 60s;
        proxy_read_timeout 60s;
    }

    # Proxy untuk MediaMTX HLS streams (opsional - jika ingin melalui Nginx)
    location /hls/ {
        proxy_pass http://localhost:8888/;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;

        # CORS headers untuk HLS
        add_header Access-Control-Allow-Origin *;
        add_header Access-Control-Allow-Methods 'GET, OPTIONS';
        add_header Cache-Control no-cache;
    }

    # Untuk file uploads
    client_max_body_size 100M;
}

```

5. Verifikasi HTTP/2 Aktif

Setelah konfigurasi selesai, verifikasi HTTP/2:

```
# Test dengan curl
curl -I --http2 -s https://your-domain.com | grep -i http

# Harus menampilkan: HTTP/2 200
```

Atau cek di browser: Developer Tools → Network → Protocol column (harus menunjukkan “h2”)

Setup SSL dengan Let’s Encrypt

1. Install Certbot

```
sudo apt install -y certbot python3-certbot-nginx
```

2. Dapatkan SSL Certificate

```
sudo certbot --nginx -d your-domain.com
```

Ikuti instruksi yang muncul.

3. Auto-Renewal

Certbot otomatis membuat cron job. Untuk test:

```
sudo certbot renew --dry-run
```

Setup MediaMTX (Media Server)

MediaMTX adalah media server yang diperlukan untuk mengkonversi RTSP/RTMP stream dari kamera CCTV ke format HLS yang dapat diputar oleh browser.

Penting: Browser tidak dapat memutar RTSP/RTMP secara langsung. MediaMTX berfungsi sebagai “penerjemah” dari RTSP/RTMP ke HLS.

1. Download dan Install MediaMTX

```
# Buat direktori untuk MediaMTX
cd /opt

# Download MediaMTX (cek versi terbaru di: https://github.com/bluenviron/mediamtx/releases)
sudo wget https://github.com/bluenviron/mediamtx/releases/download/v1.9.3/mediamtx_v1.9.3_linux_amd64.tar.gz

# Extract
sudo tar -xzf mediamtx_v1.9.3_linux_amd64.tar.gz

# Hapus file archive
sudo rm mediamtx_v1.9.3_linux_amd64.tar.gz

# Verifikasi instalasi
./mediamtx --version
```

2. Buat File Konfigurasi

```
sudo nano /opt/mediamtx.yml
```

Isi dengan konfigurasi berikut:

```
#####
# MediaMTX Configuration untuk CCTV Dashboard
#####

# Alamat server HLS (yang akan diakses browser)
hlsAddress: :8888

# Alamat server RTSP
rtspAddress: :8554

# Alamat server RTMP
rtmpAddress: :1935

# Pengaturan HLS
hlsAlwaysRemux: no
hlsSegmentCount: 3
hlsSegmentDuration: 1s
hlsPartDuration: 200ms
hlsSegmentMaxSize: 50M

# Pengaturan logging
logLevel: info
logDestinations: [stdout]

#####
# Konfigurasi Kamera (Paths)
#####

paths:
  # Template untuk semua path
  all:
    source: publisher
    sourceOnDemand: yes
    sourceOnDemandStartTimeout: 10s
    sourceOnDemandCloseAfter: 10s

#-----
# Contoh: Kamera dengan RTSP URL
#-----
# Format nama: cam1, cam2, parkir, lobby, dll

cam1:
  source: rtsp://admin:password@192.168.1.101:554/stream1
  sourceOnDemand: yes

cam2:
  source: rtsp://admin:password@192.168.1.102:554/stream1
  sourceOnDemand: yes

cam3:
  source: rtsp://admin:password@192.168.1.103:554/stream1
  sourceOnDemand: yes

cam4:
  source: rtsp://admin:password@192.168.1.104:554/stream1
  sourceOnDemand: yes

#-----
# Contoh: Sub-stream (resolusi rendah) untuk grid view
# Gunakan sub-stream untuk mengurangi bandwidth
#-----
```

```
cam1_sub:
  source: rtmp://admin:password@192.168.1.101:554/stream2
  sourceOnDemand: yes

#-----
# Contoh: Stream RTMP (untuk testing)
#-----

demo:
  source: rtmp://live.a71.ru/demo/1
  sourceOnDemand: yes

#-----
# Tambahkan kamera lainnya di bawah ini
# Sesuaikan dengan IP, username, password kamera Anda
#-----
```

3. Test MediaMTX (Manual)

```
cd /opt
sudo ./mediamtx

# Tekan Ctrl+C untuk stop
```

4. Buat Systemd Service

```
sudo nano /etc/systemd/system/mediamtx.service
```

Isi dengan:

```
[Unit]
Description=MediaMTX Media Server
Documentation=https://github.com/bluenviron/mediamtx
After=network.target

[Service]
Type=simple
User=root
ExecStart=/opt/mediamtx /opt/mediamtx.yml
Restart=always
RestartSec=5
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=mediamtx

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

5. Start MediaMTX Service

```
# Reload systemd
sudo systemctl daemon-reload

# Enable service (auto-start saat boot)
sudo systemctl enable mediamtx

# Start service
sudo systemctl start mediamtx

# Check status
sudo systemctl status mediamtx
```

6. Verifikasi MediaMTX Berjalan

```
# Check port yang digunakan
sudo netstat -tlnp | grep -E '8888|8554|1935'

# Test akses HLS (ganti cam1 dengan path yang ada)
curl -I http://localhost:8888/cam1/index.m3u8

# Check logs
sudo journalctl -u mediamtx -f
```

7. URL yang Dimasukkan di Dashboard

Setelah MediaMTX berjalan, masukkan URL HLS di dashboard:

Jenis	Format URL di Dashboard
Kamera lokal	<code>http://SERVER_IP:8888/cam1/index.m3u8</code>
Via Nginx proxy	<code>https://your-domain.com/hls/cam1/index.m3u8</code>
Test RTMP demo	<code>http://SERVER_IP:8888/demo/index.m3u8</code>

Catatan: Masukkan URL **HLS** (`http://...index.m3u8`), bukan URL RTSP langsung dari kamera.

8. Contoh Setup untuk Brand Kamera Populer

Hikvision:

```
hikvision_cam1:
  source: rtsp://admin:password@192.168.1.100:554/Streaming/Channels/101
  sourceOnDemand: yes

# Sub-stream (resolusi rendah):
hikvision_cam1_sub:
  source: rtsp://admin:password@192.168.1.100:554/Streaming/Channels/102
  sourceOnDemand: yes
```

Dahua:

```

dahua_cam1:
  source: rtsp://admin:password@192.168.1.100:554/cam/realmonitor?channel=1&subtype=0
  sourceOnDemand: yes

# Sub-stream:
dahua_cam1_sub:
  source: rtsp://admin:password@192.168.1.100:554/cam/realmonitor?channel=1&subtype=1
  sourceOnDemand: yes

```

Uniview:

```

uniview_cam1:
  source: rtsp://admin:password@192.168.1.100:554/unicast/c1/s0/live
  sourceOnDemand: yes

```

Generic ONVIF:

```

onvif_cam1:
  source: rtsp://admin:password@192.168.1.100:554/onvif1
  sourceOnDemand: yes

```

9. Tips Optimasi

Gunakan Sub-Stream untuk Grid View:

- Main stream: Resolusi tinggi untuk single view / recording
- Sub-stream: Resolusi rendah untuk grid 2x2 atau 3x3

Contoh setup dual stream per kamera:

```

# Main stream (untuk single view)
cam1_main:
  source: rtsp://admin:password@192.168.1.101:554/stream1
  sourceOnDemand: yes

# Sub stream (untuk grid view)
cam1_sub:
  source: rtsp://admin:password@192.168.1.101:554/stream2
  sourceOnDemand: yes

```

Di dashboard, gunakan:

- `http://SERVER:8888/cam1_sub/index.m3u8` untuk grid view
- `http://SERVER:8888/cam1_main/index.m3u8` untuk single camera view

10. MediaMTX Commands Berguna

```
# Restart service
sudo systemctl restart mediamtx

# Stop service
sudo systemctl stop mediamtx

# Check logs real-time
sudo journalctl -u mediamtx -f

# Edit konfigurasi (perlu restart setelah edit)
sudo nano /opt/mediamtx.yml
sudo systemctl restart mediamtx
```

Firewall Setup

```
# Enable UFW
sudo ufw enable

# Allow SSH
sudo ufw allow ssh

# Allow HTTP dan HTTPS
sudo ufw allow 80
sudo ufw allow 443

# Allow MediaMTX ports
sudo ufw allow 8888/tcp # HLS
sudo ufw allow 8554/tcp # RTSP
sudo ufw allow 1935/tcp # RTMP

# Check status
sudo ufw status
```

Troubleshooting

Database Connection Error

```
# Test koneksi database
psql -U cctv_user -h localhost -d cctv_dashboard

# Check PostgreSQL status
sudo systemctl status postgresql

# Check PostgreSQL logs
sudo tail -f /var/log/postgresql/postgresql-14-main.log
```


Aplikasi Tidak Bisa Start

```
# Check PM2 logs
pm2 logs cctv-dashboard --lines 100

# Check if port 3000 is in use
sudo netstat -tlnp | grep 3000

# Kill process yang menggunakan port 3000
sudo kill -9 $(sudo lsof -t -i:3000)
```

Nginx Error

```
# Check Nginx error logs
sudo tail -f /var/log/nginx/error.log

# Test Nginx config
sudo nginx -t
```

Prisma Issues

```
# Regenerate Prisma Client
yarn prisma generate

# Sinkronisasi schema ke database (GUNAKAN INI, BUKAN migrate deploy)
yarn prisma db push

# Reset database (HATI-HATI: Menghapus semua data!)
yarn prisma db push --force-reset

# Re-seed data
yarn prisma db seed
```

Error: “No migration found in prisma/migrations”

Project ini menggunakan `prisma db push`, bukan migrations. Gunakan:

```
# Benar:
yarn prisma db push

# Salah (jangan gunakan):
yarn prisma migrate deploy
```

Permission Issues

```
# Fix ownership
sudo chown -R $USER:$USER /var/www/cctv-dashboard

# Fix permissions
chmod -R 755 /var/www/cctv-dashboard
```

MediaMTX Issues

Stream tidak muncul / Loading terus:

```
# Check MediaMTX running
sudo systemctl status mediamtx

# Check logs
sudo journalctl -u mediamtx -f

# Restart MediaMTX
sudo systemctl restart mediamtx
```

Error “source not found” atau “connection refused”:

```
# Test RTSP URL langsung dengan ffprobe
ffprobe rtsp://admin:password@192.168.1.100:554/stream1

# Pastikan IP kamera dapat diakses dari server
ping 192.168.1.100

# Check firewall kamera (port 554 harus terbuka)
```

HLS tidak bisa diakses:

```
# Test akses HLS
curl -v http://localhost:8888/cam1/index.m3u8

# Check port 8888 listen
sudo netstat -tlnp | grep 8888

# Pastikan firewall allow port 8888
sudo ufw allow 8888/tcp
```

Stream lag / buffering:

```
# Edit mediamtx.yml - kurangi segment duration
hlsSegmentDuration: 500ms
hlsPartDuration: 100ms
hlsSegmentCount: 2
```

Update Aplikasi

Untuk update aplikasi ke versi terbaru:

```
cd /var/www/cctv-dashboard

# Stop aplikasi
pm2 stop cctv-dashboard

# Pull changes (jika menggunakan git)
git pull origin main

# Install dependencies baru
yarn install

# Regenerate Prisma (jika ada perubahan schema)
yarn prisma generate

# Push schema changes (jika ada)
yarn prisma db push

# Build ulang
yarn build

# Restart aplikasi
pm2 restart cctv-dashboard
```

Backup Database

Backup

```
# Backup database
pg_dump -U cctv_user -h localhost cctv_dashboard > backup_$(date +%Y%m%d).sql
```

Restore

```
# Restore database
psql -U cctv_user -h localhost cctv_dashboard < backup_20240101.sql
```

Quick Start Script

Simpan script ini sebagai `setup.sh` untuk instalasi cepat:

```
#!/bin/bash
set -e

echo "=== CCTV Dashboard Setup Script ==="

# Update system
sudo apt update && sudo apt upgrade -y

# Install Node.js 18
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs

# Install Yarn
sudo npm install -g yarn

# Install PostgreSQL
sudo apt install -y postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Install Nginx
sudo apt install -y nginx

# Install PM2
sudo npm install -g pm2

# Install build tools
sudo apt install -y build-essential git curl wget

# Install MediaMTX
echo "=== Installing MediaMTX ==="
cd /opt
sudo wget -q https://github.com/bluenvron/mediamtx/releases/download/v1.9.3/mediamtx_v1.9.3_linux_amd64.tar.gz
sudo tar -xzf mediamtx_v1.9.3_linux_amd64.tar.gz
sudo rm mediamtx_v1.9.3_linux_amd64.tar.gz

# Create MediaMTX systemd service
sudo tee /etc/systemd/system/mediamtx.service > /dev/null <<EOF
[Unit]
Description=MediaMTX Media Server
After=network.target

[Service]
Type=simple
User=root
ExecStart=/opt/mediamtx /opt/mediamtx.yml
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable mediamtx

# Configure firewall
sudo ufw allow ssh
sudo ufw allow 80
sudo ufw allow 443
sudo ufw allow 8888/tcp # HLS
```

```

sudo ufw allow 8554/tcp # RTSP
sudo ufw allow 1935/tcp # RTMP

echo "=== Dependencies installed! ==="
echo ""
echo "Next steps:"
echo "1. Setup PostgreSQL database (buat user dan database)"
echo "2. Configure .env file (DATABASE_URL, NEXTAUTH_SECRET, NEXTAUTH_URL)"
echo "3. Run: yarn install"
echo "4. Run: yarn prisma generate"
echo "5. Run: yarn prisma db push (JANGAN gunakan migrate deploy)"
echo "6. Run: yarn prisma db seed (opsional, untuk data demo)"
echo "7. Run: yarn build"
echo "8. Run: pm2 start ecosystem.config.js"
echo ""
echo "9. Configure MediaMTX:"
echo "  - Edit /opt/mediamtx.yml dengan RTSP URL kamera Anda"
echo "  - Run: sudo systemctl start mediamtx"
echo ""
echo "10. Setup SSL dan HTTP/2:"
echo "  - Run: sudo certbot --nginx -d your-domain.com"
echo "  - Update Nginx config untuk HTTP/2 (lihat DEPLOYMENT.md)"

```

Jalankan dengan:

```

chmod +x setup.sh
./setup.sh

```

Kontak & Bantuan

Jika mengalami masalah:

1. Check dokumentasi Prisma: <https://www.prisma.io/docs>
 2. Check dokumentasi Next.js: <https://nextjs.org/docs>
 3. Check dokumentasi NextAuth: <https://next-auth.js.org/getting-started/introduction>
-

Selamat! Aplikasi CCTV Dashboard Anda sekarang sudah berjalan di production. 🎉