

## SISTEM ARSIP DIGITAL

Dinas Sosial

# MANUAL INSTALASI

*Panduan Konfigurasi & Instalasi dari Nol*

v1.0.0 | 2025

# PRASYARAT SISTEM

---

Sebelum memulai instalasi, pastikan sistem komputer Anda memenuhi persyaratan berikut:

## Perangkat Keras Minimum

Komponen	Minimum	Rekomendasi
Processor	Dual Core 2 GHz	Quad Core 2.5 GHz+
RAM	2 GB	4 GB+
Storage	1 GB free	5 GB+ free
OS	Windows 10 / Ubuntu 20.04 / macOS 11	Versi terbaru

## Perangkat Lunak yang Harus Diinstall

- Node.js**: Versi 20 atau lebih baru (wajib). Download dari <https://nodejs.org>
- Git**: Untuk clone repository (opsional jika menggunakan ZIP)
- Text Editor**: VS Code direkomendasikan untuk mengedit konfigurasi

*Cek versi Node.js: Buka terminal/command prompt, ketik: node --version. Harus menampilkan v20.x.x atau lebih tinggi.*

---

# LANGKAH 1: PERSIAPAN PROJECT

---

## 1.1 Mendapatkan Source Code

Opsi A - Menggunakan Git Clone:

```
git clone <URL_REPOSITORY> arsip-digital  
cd arsip-digital
```

Opsi B - Menggunakan File ZIP:

- Download file ZIP project dari sumber yang diberikan
- Ekstrak ZIP ke folder yang diinginkan (contoh: C:\Projects\arsip-digital)
- Buka terminal/command prompt dan masuk ke folder tersebut

```
cd C:\Projects\arsip-digital
```

---

# LANGKAH 2: INSTALASI DEPENDENCIES

---

Jalankan perintah berikut di dalam folder project untuk menginstall semua library yang dibutuhkan:

```
npm install
```

Proses ini akan mengunduh semua package yang tercantum di package.json. Pastikan koneksi internet stabil. Waktu yang dibutuhkan: 2-10 menit tergantung kecepatan internet.

**Jika muncul error ERESOLVE, coba: npm install --legacy-peer-deps**

## LANGKAH 3: KONFIGURASI ENVIRONMENT

### 3.1 Membuat File .env

File .env berisi konfigurasi yang berbeda antara development dan production. Buat file .env di root folder project:

Untuk Development (jalankan di komputer lokal):

```
DATABASE_URL=file:./data/arsip_dinsos.db  
NODE_ENV=development  
USE_HTTPS=false
```

Untuk Production (server lokal/jaringan internal):

```
DATABASE_URL=file:/data/arsip_dinsos.db  
NODE_ENV=production  
USE_HTTPS=false
```

### 3.2 Penjelasan Variabel Konfigurasi

Variabel	Nilai	Keterangan
DATABASE_URL	file:./data/arsip_dinsos.db	Path ke file database SQLite
NODE_ENV	development / production	Mode aplikasi
USE_HTTPS	true / false	Set true HANYA jika menggunakan HTTPS/SSL

## LANGKAH 4: SETUP DATABASE

### 4.1 Membuat Folder Data

Pastikan folder data/ ada di root project:

```
mkdir data
```

### 4.2 Menggunakan Database Backup (Jika Ada)

Jika Anda memiliki file backup database dari sistem yang sudah berjalan sebelumnya:

4. Salin file .db backup ke folder data/
5. Rename file menjadi arsip\_dinsos.db

```
cp /path/to/backup.db ./data/arsip_dinsos.db
```

## 4.3 Database Fresh (Instalasi Baru)

Jika ini instalasi baru, jalankan migrasi untuk membuat tabel-tabel yang diperlukan:

```
npx drizzle-kit push
```

Perintah ini akan membuat semua tabel berdasarkan schema yang didefinisikan di src/db/schema.ts.

## 4.4 Membuat Akun Admin Awal

Setelah database dibuat, Anda perlu membuat akun admin pertama. Buat file seed.js sementara:

```
// seed.js - Jalankan sekali untuk buat admin
const Database = require('better-sqlite3');
const bcrypt = require('bcrypt');
const db = new Database('./data/arsip_dinsos.db');
const hash = bcrypt.hashSync('admin123', 10);
db.prepare("INSERT INTO users (username, password, role) VALUES (?, ?, ?")
    .run('admin', hash);
console.log('Admin berhasil dibuat');
```

Jalankan:

```
node seed.js
```

**PENTING:** Segera ubah password admin setelah login pertama kali melalui menu Pengguna!

---

# LANGKAH 5: MENJALANKAN APLIKASI

---

## 5.1 Mode Development

Untuk pengembangan dan testing:

```
npm run dev
```

Aplikasi akan berjalan di <http://localhost:3000>. Mode ini memiliki hot-reload (perubahan file otomatis terdeteksi).

## 5.2 Mode Production

Untuk penggunaan nyata/sehari-hari, build terlebih dahulu:

```
npm run build
```

Setelah build selesai, jalankan:

```
npm start
```

Aplikasi berjalan di <http://localhost:3000> dalam mode production yang lebih cepat dan optimal.

### 5.3 Menjalankan di Port Berbeda

```
PORT=8080 npm start
```

Ganti 8080 dengan port yang diinginkan.

---

## LANGKAH 6: KONFIGURASI NEXT.JS

---

### 6.1 File next.config.ts

Untuk production build dengan Docker (standalone mode), pastikan file next.config.ts memiliki konfigurasi output standalone:

```
// next.config.ts
const nextConfig = {
  output: 'standalone',
};
export default nextConfig;
```

*output: 'standalone' diperlukan untuk Docker deployment. Tanpa ini, build Docker tidak akan bekerja dengan benar.*

---

## LANGKAH 7: AKSES DARI JARINGAN LOKAL

---

Secara default Next.js hanya dapat diakses dari komputer yang menjalankan (localhost). Untuk mengizinkan komputer lain di jaringan yang sama mengakses sistem:

```
npm start -- -H 0.0.0.0
```

Atau edit package.json scripts:

```
"start": "next start -H 0.0.0.0"
```

Kemudian komputer lain di jaringan bisa akses menggunakan IP komputer server, contoh:  
<http://192.168.1.10:3000>

---

## TROUBLESHOOTING INSTALASI

---

Masalah	Kemungkinan Penyebab	Solusi
npm install gagal	Versi Node.js terlalu lama	Update Node.js ke v20+
Database error saat start	Folder data/ belum ada	Buat folder data/: mkdir data

Masalah	Kemungkinan Penyebab	Solusi
Error better-sqlite3	Native module perlu dikompilasi	npm install --build-from-source
Port 3000 sudah terpakai	Aplikasi lain menggunakan port	Gunakan port lain: PORT=3001 npm start
Build gagal	Error TypeScript	Cek log error, perbaiki kode