

SISTEM ARSIP DIGITAL

Dinas Sosial

MANUAL VERCCEL

Deploy & Manajemen di Vercel

v1.0.0 | 2025

TENTANG VERCEL DEPLOYMENT

Vercel adalah platform cloud yang secara khusus dioptimalkan untuk Next.js. Deployment ke Vercel memungkinkan aplikasi diakses dari internet (bukan hanya jaringan lokal) dengan SSL/HTTPS otomatis, CDN global, dan domain gratis (.vercel.app).

CATATAN PENTING: Karena aplikasi menggunakan SQLite (database file lokal), ada batasan khusus di Vercel. SQLite tidak cocok untuk production di Vercel karena filesystem Vercel bersifat read-only dan ephemeral (data hilang saat redeploy). Solusi yang direkomendasikan dijelaskan di Bagian 3.

1. PRASYARAT

- Akun GitHub / GitLab / Bitbucket (untuk repository)
- Akun Vercel - daftar gratis di <https://vercel.com>
- Source code project sudah di-push ke repository Git

Keterbatasan Vercel dengan SQLite

SQLite menyimpan data sebagai file di disk. Di Vercel:

- Filesystem bersifat read-only, kecuali folder /tmp
- Folder /tmp hanya tersedia untuk satu function invocation, tidak persisten
- Data akan hilang setiap kali project di-redeploy

Oleh karena itu, untuk deployment Vercel yang sesungguhnya (data persisten), ada 3 opsi:

| Opsi | Deskripsi | Biaya |
|----------------------|--|------------------------|
| Turso (SQLite cloud) | SQLite yang disimpan di cloud, compatible dengan kode existing | Gratis (tier terbatas) |
| Neon (PostgreSQL) | Migrasi database ke PostgreSQL cloud | Gratis (tier terbatas) |
| PlanetScale (MySQL) | Migrasi database ke MySQL cloud | Gratis (tier terbatas) |

Rekomendasi termudah: Gunakan Turso karena interface SQLite-nya hampir identik, perubahan kode minimal.

2. DEPLOY VERCEL (MODE DEMO/TESTING)

Untuk testing atau demo (data tidak persisten, reset setiap deploy):

2.1 Push Code ke GitHub

1. Buat repository baru di GitHub
2. Di folder project, inisialisasi Git dan push:

```
git init
git add .
git commit -m 'Initial commit'
git remote add origin https://github.com/username/arsip-digital.git
git push -u origin main
```

2.2 Import ke Vercel

3. Login ke <https://vercel.com>
4. Klik New Project
5. Pilih Import Git Repository
6. Pilih repository arsip-digital
7. Klik Import

2.3 Konfigurasi Project

Pada halaman konfigurasi deployment:

- **Framework Preset:** Vercel akan otomatis mendeteksi Next.js
- **Root Directory:** Biarkan default (kosong) jika tidak dalam subfolder
- **Build Command:** next build (default)
- **Output Directory:** .next (default)

2.4 Environment Variables

Tambahkan environment variables yang diperlukan:

8. Klik bagian Environment Variables
9. Tambahkan variabel berikut:

| Key | Value | Environment |
|--------------|---------------------------|-------------|
| DATABASE_URL | file:/tmp/arsip_dinsos.db | All |
| NODE_ENV | production | Production |
| USE_HTTPS | true | Production |

10. Klik Deploy

Catatan: Dengan DATABASE_URL=/tmp/..., data TIDAK akan persisten. Gunakan hanya untuk demo/testing.

3. DEPLOY VERCEL (PRODUKSI DENGAN TURSO)

Untuk deployment produksi dengan data persisten menggunakan Turso (SQLite cloud):

3.1 Buat Database Turso

11. Daftar/Login di <https://turso.tech>

12. Buat database baru:

```
turso db create arsip-digital
```

13. Dapatkan URL database:

```
turso db show arsip-digital
```

14. Buat authentication token:

```
turso db tokens create arsip-digital
```

15. Catat URL database (format: libsql://...) dan token

3.2 Migrasi Kode ke Turso

Modifikasi src/db/index.ts untuk menggunakan Turso:

16. Install library Turso:

```
npm install @libsql/client drizzle-orm
```

17. Update src/db/index.ts:

```
import { createClient } from '@libsql/client';
import { drizzle } from 'drizzle-orm/libsql';

const client = createClient({
  url: process.env.TURSO_DATABASE_URL!,
  authToken: process.env.TURSO_AUTH_TOKEN,
});

export const db = drizzle(client, { schema });
```

Catatan: Perubahan ini mempengaruhi semua query database. Pastikan semua raw SQL query menggunakan syntax yang kompatibel dengan LibSQL (Turso).

3.3 Konfigurasi Environment Variables di Vercel

18. Di Vercel dashboard, buka Project Settings > Environment Variables

19. Tambahkan:

| Key | Value |
|--------------------|---|
| TURSO_DATABASE_URL | libsql://your-db-name-username.turso.io |
| TURSO_AUTH_TOKEN | your-auth-token-from-turso |
| NODE_ENV | production |
| USE_HTTPS | true |

20. Klik Save

3.4 Redeploy

```
git add .
git commit -m 'Migrate to Turso for Vercel production'
git push
```

Vercel akan otomatis mendeteksi push dan melakukan redeploy.

4. MANAJEMEN DEPLOYMENT VERCEL

4.1 Dashboard Vercel

Setelah deploy, Anda dapat mengelola deployment melalui <https://vercel.com/dashboard>:

- **Deployments:** Lihat riwayat deployment, rollback ke versi sebelumnya
- **Analytics:** Monitor traffic dan performance aplikasi
- **Logs:** Lihat runtime logs untuk debugging
- **Environment Variables:** Kelola konfigurasi
- **Domains:** Tambah custom domain

4.2 Custom Domain

21. Di Vercel dashboard, buka project Anda
22. Masuk ke Settings > Domains
23. Klik Add Domain
24. Masukkan domain (contoh: arsip.dinsos-kota.go.id)
25. Ikuti instruksi untuk mengkonfigurasi DNS di registrar domain Anda
26. Vercel akan otomatis menyediakan SSL certificate (HTTPS)

4.3 Rollback Deployment

27. Di Vercel dashboard, buka tab Deployments
28. Cari deployment sebelumnya yang ingin dipulihkan
29. Klik tombol tiga titik (...) di sebelah deployment tersebut
30. Pilih Promote to Production

4.4 Variabel Environment per Stage

Vercel mendukung tiga environment yang berbeda:

| Environment | Keterangan | URL |
|-------------|-------------------------------|-------------------------|
| Production | Branch main/master, URL utama | your-project.vercel.app |

| Environment | Keterangan | URL |
|-------------|------------------------------|--------------------|
| Preview | Pull request dan branch lain | Auto-generated URL |
| Development | Untuk pengembangan lokal | localhost:3000 |

5. CI/CD OTOMATIS

Setelah repository terhubung ke Vercel, setiap push ke GitHub akan otomatis mentrigger deployment baru:

- Push ke branch main -> Deploy ke Production
- Push ke branch lain -> Deploy ke Preview (URL unik)
- Pull Request -> Preview deployment dengan link yang dapat dikomentar

Disable Auto-Deploy

Jika tidak ingin auto-deploy, atur di Vercel Settings > Git > Deploy Hooks atau ubah branch settings:

31. Buka Project Settings > Git
32. Ubah Production Branch ke branch tertentu yang Anda kontrol
33. Atau disable automatic deployments dan trigger manual via Vercel CLI

6. MONITORING & LOGS

Melihat Logs

34. Di Vercel dashboard, buka project
35. Klik tab Functions atau Logs
36. Pilih deployment yang ingin dimonitor
37. Log dari console.log dan error akan tampil di sini

Vercel CLI

Untuk monitoring dan deployment via terminal:

```
npm install -g vercel  
vercel login  
vercel logs your-project-name  
vercel --prod          # Deploy ke production
```

RINGKASAN PERBANDINGAN DEPLOYMENT

| Aspek | Server Lokal (NPM) | Docker Lokal | Vercel |
|-------------------|----------------------|------------------------|------------------------|
| Aksesibilitas | Jaringan lokal | Jaringan lokal | Internet global |
| Setup | Sedang | Mudah (setelah Docker) | Sangat mudah |
| Persistensi Data | Penuh (SQLite lokal) | Penuh (volume) | Butuh Turso/cloud DB |
| Performa | Baik | Baik | Sangat baik (CDN) |
| SSL/HTTPS | Manual | Manual | Otomatis |
| Biaya | Gratis | Gratis | Gratis (tier terbatas) |
| Rekomendasi Untuk | Dev, kantor tertutup | Prod kantor lokal | Demo, akses publik |