RANGKUMAN LAPORAN — MINI ERP PENGIRIMAN
STACK: VUE 3 + VITE (FRONTEND) · VERCEL FUNCTIONS (API) · **NEON (POSTGRESQL)* FOTO/POD)**
TANGGAL: 12 NOVEMBER 2025

---

1) Ringkasan Solusi (final)

- **Frontend**: Vue 3 + Vite, dideploy **statis di Vercel**.
- **Backend/API**: Node 18 (Hono/Express) sebagai **Vercel Serverless Functions** di folder `/api`.
- **Database**: **Neon (PostgreSQL)** untuk seluruh data transaksional.
- **Storage**: **Vercel Blob** untuk foto POD & dokumen (surat jalan scan, dll). *Neon tidak menyediaka object storage—jadi file disimpan di Vercel Blob, metadatanya disimpan di Neon.*
- **POD**: halaman publik bertoken (sekali pakai) + PIN sederhana; tanda tangan digital + foto → uploa Blob.
- **Dokumen**: PDF Surat Jalan & Invoice dirender di API (serverless).

Target biaya tetap **< Rp150rb/bln** (Vercel Hobby + Neon Free + domain opsional).

---

2) Arsitektur

```
[Vue 3 + Vite (static @Vercel)]
        |
        +--> [/api/* Vercel Functions (Node 18, Hono/Express)]
                |
                +--> Neon (PostgreSQL) — data & metadata file
                +--> Vercel Blob — penyimpanan foto/ttd dokumen
```

- **Keamanan**: JWT untuk admin; token sekali pakai untuk POD; semua upload file **via API** (server-side) agar kunci/credential tidak bocor.
- **Ukuran file**: batasi ~5MB/foto (kompres di browser).

---

3) Skema Data Inti (Neon)

```sql
CREATE TABLE customers (
  id BIGSERIAL PRIMARY KEY,
  name TEXT NOT NULL,
  phone TEXT,
  address TEXT,
  tax_id TEXT,
  created_at TIMESTAMPTZ DEFAULT now()
);

CREATE TABLE shipments (
  id BIGSERIAL PRIMARY KEY,
  customer_id BIGINT REFERENCES customers(id),
  origin TEXT NOT NULL,
  destination TEXT NOT NULL,
  eta DATE,
  status TEXT NOT NULL DEFAULT 'DRAFT',
  total_colli INT DEFAULT 0,
  public_code TEXT UNIQUE,
  created_at TIMESTAMPTZ DEFAULT now()
);

CREATE TABLE colli (
  id BIGSERIAL PRIMARY KEY,
  shipment_id BIGINT REFERENCES shipments(id) ON DELETE CASCADE,
  code TEXT UNIQUE,
  weight NUMERIC(10,2),
  status TEXT DEFAULT 'READY'
);

CREATE TABLE trips (
  id BIGSERIAL PRIMARY KEY,
  carrier_name TEXT,
  vehicle_plate TEXT,
  driver_name TEXT,
  driver_phone TEXT,
  depart_at TIMESTAMPTZ,
  arrive_est TIMESTAMPTZ
);

CREATE TABLE trip_items (
  id BIGSERIAL PRIMARY KEY,
  trip_id BIGINT REFERENCES trips(id) ON DELETE CASCADE,
  shipment_id BIGINT REFERENCES shipments(id) ON DELETE CASCADE
);

CREATE TABLE delivery_tokens (
  id BIGSERIAL PRIMARY KEY,
  shipment_id BIGINT REFERENCES shipments(id) ON DELETE CASCADE,
  token TEXT UNIQUE NOT NULL,
  expires_at TIMESTAMPTZ,
  used_at TIMESTAMPTZ
);

CREATE TABLE pod (
  id BIGSERIAL PRIMARY KEY,
  shipment_id BIGINT REFERENCES shipments(id) ON DELETE CASCADE,
  method TEXT NOT NULL, -- 'pin' | 'email_otp' | 'photo_only'
  signed_name TEXT,
  signed_at TIMESTAMPTZ,
  gps_lat DOUBLE PRECISION,
  gps_lng DOUBLE PRECISION,
  photos JSONB DEFAULT '[]'::jsonb -- [{url, pathname, size, type, sha256}]
```

```
);

CREATE TABLE events (
  id BIGSERIAL PRIMARY KEY,
  entity_type TEXT NOT NULL,
  entity_id BIGINT NOT NULL,
  event_type TEXT NOT NULL,
  payload_json JSONB,
  created_by BIGINT,
  created_at TIMESTAMPTZ DEFAULT now()
);

CREATE TABLE users (
  id BIGSERIAL PRIMARY KEY,
  email TEXT UNIQUE NOT NULL,
  name TEXT,
  password_hash TEXT NOT NULL,
  role TEXT DEFAULT 'staff',
  created_at TIMESTAMPTZ DEFAULT now()
);
```

Index penting: `shipments(status, destination)`, `colli(code)`, `events(entity_type, entity_id, created_at DESC)`, `delivery_tokens(token)`.

---

4) Deploy (Vercel + Neon)
Struktur proyek:

```
/api                    # Vercel Functions
/src                    # Vue 3
/public
vercel.json
package.json
index.html
```

vercel.json

```json
{
  "buildCommand": "pnpm build",
  "outputDirectory": "dist",
  "functions": {
    "api/**/*.ts": { "runtime": "nodejs18.x" }
  },
  "routes": [
    { "src": "^/api/(.*)$", "dest": "/api/$1" },
    { "src": "^(?!/api/).*", "dest": "/" }
  ]
}
```

```

**ENV (Vercel Project)**
```
DATABASE_URL=postgres://user:pass@ep-xxxxx.neon.tech/neondb
JWT_SECRET=supersecret
BLOB_READ_WRITE_TOKEN=... # Storage → Blob → Generate RW token

---

5) Upload ke **Vercel Blob** (aman & sederhana)

Server (Vercel Function, Node — file ≤5MB):
```ts
// /api/blob/upload.ts
import { put } from '@vercel/blob';
import crypto from 'crypto';

export const config = { runtime: 'nodejs18.x' };

export default async function handler(req, res) {
  if (req.method !== 'POST') return res.status(405).end();

const ext = (req.query.ext || 'jpg').toString();
  const key = `erp/${new Date().toISOString().slice(0,7).replace('-', '/')}/${crypto.randomUUID()}.${ext}`;

const arrayBuffer = await req.arrayBuffer();
  const blob = await put(key, Buffer.from(arrayBuffer), {
    access: 'private',
    token: process.env.BLOB_READ_WRITE_TOKEN,
    contentType: req.headers['content-type'] || 'image/jpeg'
  });

// TODO: simpan metadata ke Neon
  res.json({ url: blob.url, pathname: blob.pathname, size: blob.size });
}
```

Client (Vue) — upload:
```ts
async function uploadPOD(file: File) {
  const res = await fetch('/api/blob/upload?ext=' + (file.name.split('.').pop() || 'jpg'), {
    method: 'POST',
    headers: { 'Content-Type': file.type || 'application/octet-stream' },
    body: await file.arrayBuffer()
  });
  const data = await res.json();
  if (!res.ok) throw new Error(data.error || 'Upload gagal');
  return data; // { url, pathname, size }
}
```

Akses file (signed GET)
Buat route server yang validasi kepemilikan dan mengembalikan URL akses sementara (lihat dokumentasi Vercel Blob). Simpan **pathname** di Neon agar stabil.

---

6) AI-KICKOFF (untuk VS Code AI)

Tasks

1. **Buat `vercel.json` seperti di atas.**
2. **Implement `/api/blob/upload.ts` dengan `@vercel/blob` `put()` (private).**
3. **Buat `src/pages/PodUpload.vue` (kompres gambar, size guard 5MB, progress bar).**
4. **Tulis service Neon untuk menyimpan metadata POD (`photos`).**
5. **Buat endpoint `POST /pod/:token/submit` yang mengikat foto ke shipment + menutup token.**

Acceptance Criteria
- Deploy sukses di Vercel, Neon connected.
- Upload foto POD <5MB berhasil → tersimpan di **Vercel Blob** (private) dan metadata tersimpan di Neon.
- Halaman admin menampilkan foto POD via signed link.

---

7) Tips

- **Kompres di browser ke 1280px (quality 0.7).**
- **Validasi mime: `image/jpeg,image/png,application/pdf`.**
- **Batasi jumlah foto per POD (max 3).**
- **Retensi: arsip >12 bulan bila dibutuhkan.**