

////////////////////////////////////

Nama: Mohamad Basuki

Telp/Wa:+6281295332140

Test membuat Aplikasi Kecil

////////////////////////////////////

Daftar ISI

- A. Panduan Install
- B. Running Application and test
- C. Struktur Database
- D. Upload GitHub
- E. Rekaman Aplikasi
- F. Catatan

A. **Panduan Install**

1. **Install**

install node js versi v24.12.0

- <https://nodejs.org/en/download>

Instll sql lite 3.13.1

- <https://sqlitebrowser.org/dl/>

Install Git (Required) version 2.52.

- <https://git-scm.com/>

2. **Check hasil instalasi**

check node

C:\Users\laptop41>node -v

v24.12.0

check npm

C:\Users\laptop41>npm -v

11.7.0

check git

C:\Windows\System32>git --version

git version 2.52.0.windows.1

3. **Create Folder untuk Project & install Library**

- Buat folder di C:\admin-system jika sudah jadi sorot klik kanan pilih Open in Terminal

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!
<https://aka.ms/PSWindows>

PS C:\admin-system>

- **Pada Command ketik npm init -y**

```
{
  "name": "admin-system",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

- **menghubungkan Node.js to SQLite and Express, ketik command dibawah ini:**
npm install express ejs sqlite3 axios dotenv

- **buat konfigurasi file menggunakan notepad++**

- nama file .env dan isi filenya

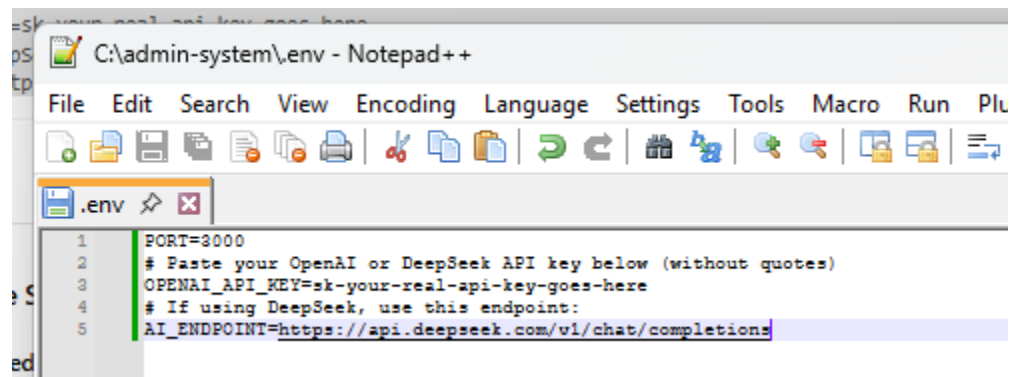
PORT=3000

Paste your OpenAI or DeepSeek API key below (without quotes)

OPENAI_API_KEY=sk-your-real-api-key-goes-here

If using DeepSeek, use this endpoint:

AI_ENDPOINT=https://api.deepseek.com/v1/chat/completions



-

- nama file server.js dengan kode dibawah ini

```
=====
require('dotenv').config();
const express = require('express');
const sqlite3 = require('sqlite3').verbose();
const axios = require('axios');
const path = require('path');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(express.json());
app.use(express.static('public')); // Serves CSS/JS
app.set('view engine', 'ejs');    // Serves HTML files

// --- DATABASE SETUP ---
const db = new sqlite3.Database('./database.db');

db.serialize(() => {
  // 1. Products Table
  db.run(`CREATE TABLE IF NOT EXISTS products (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT,
    price REAL
  )`);

  // 2. Stock Table
  db.run(`CREATE TABLE IF NOT EXISTS stock (
    product_id INTEGER,
    quantity INTEGER,
    FOREIGN KEY(product_id) REFERENCES products(id)
  )`);

  // 3. Purchases Table
  db.run(`CREATE TABLE IF NOT EXISTS purchases (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    date TEXT,
    customer TEXT,
    product_id INTEGER,
    quantity INTEGER,
    total REAL,`
```

```

    status TEXT DEFAULT 'Active'
  ));

// 4. Seed Data (Insert sample products if empty)
db.get("SELECT count(*) as count FROM products", (err, row) => {
  if (row.count === 0) {
    const products = [
      { name: "Wireless Mouse", price: 25.00, stock: 50 },
      { name: "Mechanical Keyboard", price: 85.00, stock: 30 },
      { name: "27-inch Monitor", price: 150.00, stock: 15 },
      { name: "USB-C Hub", price: 40.00, stock: 100 },
      { name: "Webcam 1080p", price: 60.00, stock: 25 },
      { name: "Headphones", price: 120.00, stock: 20 },
      { name: "Laptop Stand", price: 35.00, stock: 45 },
      { name: "External SSD 1TB", price: 110.00, stock: 60 },
      { name: "Ergonomic Chair", price: 200.00, stock: 10 },
      { name: "Smart Lamp", price: 45.00, stock: 35 }
    ];

    products.forEach(p => {
      db.run("INSERT INTO products (name, price) VALUES (?, ?)", [p.name,
p.price], function(err) {
        if (!err) {
          db.run("INSERT INTO stock (product_id, quantity) VALUES (?, ?)",
[this.lastID, p.stock]);
        }
      });
    });
    console.log("Database seeded with 10 products.");
  }
});

// --- ROUTES ---

// Render Dashboard
app.get('/', (req, res) => {
  res.render('index');
});

// API: Stats
app.get('/api/stats', (req, res) => {
  const sql = `
    SELECT

```

```

        (SELECT COUNT(*) FROM products) as totalProducts,
        (SELECT COUNT(*) FROM purchases WHERE status='Active') as activeOrders,
        (SELECT COALESCE(SUM(total), 0) FROM purchases WHERE status='Active') as
revenue
    `;
    db.get(sql, [], (err, row) => {
        if (err) return res.status(500).json({ error: err.message });
        res.json(row);
    });
});

// API: Get Products
app.get('/api/products', (req, res) => {
    const sql = `SELECT p.id, p.name, p.price, s.quantity as stock FROM products p
JOIN stock s ON p.id = s.product_id`;
    db.all(sql, [], (err, rows) => {
        if (err) return res.status(500).json({ error: err.message });
        res.json(rows);
    });
});

// API: Get Purchases
app.get('/api/purchases', (req, res) => {
    const sql = `SELECT pur.id, pur.date, pur.customer, pur.quantity, pur.total,
pur.status, p.name as productName
FROM purchases pur JOIN products p ON pur.product_id = p.id ORDER BY
pur.id DESC`;
    db.all(sql, [], (err, rows) => {
        if (err) return res.status(500).json({ error: err.message });
        res.json(rows);
    });
});

// API: Create Purchase
app.post('/api/purchase', (req, res) => {
    const { customer, productId, quantity } = req.body;

    db.get("SELECT quantity FROM stock WHERE product_id = ?", [productId], (err,
stockRow) => {
        if (err || !stockRow) return res.status(500).json({ error: "Product not found" });
        if (stockRow.quantity < quantity) return res.status(400).json({ error: "Insufficient
stock" });
    });
});

```

```

    db.get("SELECT price FROM products WHERE id = ?", [productId], (err, prodRow)
=> {
    const total = prodRow.price * quantity;
    const date = new Date().toISOString().split('T')[0];

    db.run("INSERT INTO purchases (date, customer, product_id, quantity, total)
VALUES (?, ?, ?, ?, ?)",
    [date, customer, productId, quantity, total], function(err) {
    if (err) return res.status(500).json({ error: err.message });

    db.run("UPDATE stock SET quantity = quantity - ? WHERE product_id = ?",
[quantity, productId], (err) => {
    if (err) return res.status(500).json({ error: err.message });
    res.json({ message: "Purchase created", id: this.lastID });
    });
    });
    });
    });

// API: Cancel Purchase
app.post('/api/cancel/:id', (req, res) => {
    const id = req.params.id;
    db.get("SELECT * FROM purchases WHERE id = ?", [id], (err, purchase) => {
    if (err || !purchase) return res.status(404).json({ error: "Purchase not found" });
    if (purchase.status === 'Cancelled') return res.status(400).json({ error: "Already
cancelled" });

    db.run("UPDATE purchases SET status = 'Cancelled' WHERE id = ?", [id], (err) => {
    if (err) return res.status(500).json({ error: err.message });
    db.run("UPDATE stock SET quantity = quantity + ? WHERE product_id = ?",
[purchase.quantity, purchase.product_id], (err) => {
    if (err) return res.status(500).json({ error: err.message });
    res.json({ message: "Purchase cancelled" });
    });
    });
    });
    });

// API: Chat
app.post('/api/chat', async (req, res) => {
    const userMessage = req.body.message;
    const systemPrompt = "You are a helpful Admin Assistant. Answer concisely.";

```

```

try {
  const response = await axios.post(process.env.AI_ENDPOINT, {
    model: "deepseek-chat", // or "gpt-3.5-turbo"
    messages: [
      { role: "system", content: systemPrompt },
      { role: "user", content: userMessage }
    ]
  }, {
    headers: { 'Authorization': `Bearer ${process.env.OPENAI_API_KEY}` }
  });
  res.json({ reply: response.data.choices[0].message.content });
} catch (error) {
  console.error("AI Error:", error.response ? error.response.data : error.message);
  res.json({ reply: "Sorry, AI connection failed. Check API Key." });
}
});

app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
=====

```

- buat folder public dan views didalam folder admin-system.
- didalam folder public buat file **styles.css** isi kode dibawah ini.

```

=====
:root {
  --primary: #4f46e5;
  --bg-body: #f3f4f6;
  --bg-card: #ffffff;
  --text-main: #111827;
  --border: #e5e7eb;
  --danger: #ef4444;
  --success: #10b981;
}
* { box-sizing: border-box; margin: 0; padding: 0; font-family: sans-serif; }
body { background-color: var(--bg-body); color: var(--text-main); height: 100vh; display: flex; overflow: hidden; }
aside { width: 250px; background: var(--bg-card); border-right: 1px solid var(--border); padding: 1.5rem; }
main { flex: 1; overflow-y: auto; padding: 2rem; }
.section { display: none; }
.section.active { display: block; }
.card { background: var(--bg-card); padding: 1.5rem; border-radius: 12px; border: 1px solid var(--border); margin-bottom: 1.5rem; }

```

```

.grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap:
1.5rem; }
input, select { width: 100%; padding: 0.6rem; margin-bottom: 1rem; border: 1px solid
var(--border); border-radius: 6px; }
button { padding: 0.6rem 1.2rem; border: none; border-radius: 6px; cursor: pointer;
background: var(--primary); color: white; }
.btn-danger { background: var(--danger); padding: 0.4rem 0.8rem; font-size: 0.85rem; }
table { width: 100%; border-collapse: collapse; margin-top: 1rem; }
th, td { padding: 1rem; text-align: left; border-bottom: 1px solid var(--border); }
.chat-container { height: 60vh; display: flex; flex-direction: column; border: 1px solid var(-
-border); border-radius: 12px; background: white; }
.chat-messages { flex: 1; padding: 1rem; overflow-y: auto; }
.message { margin-bottom: 10px; padding: 10px; border-radius: 8px; max-width: 70%; }
.msg-user { margin-left: auto; background: var(--primary); color: white; }
.msg-ai { margin-right: auto; background: #f3f4f6; color: black; }
.toast { position: fixed; bottom: 20px; right: 20px; padding: 1rem; background: #333;
color: white; border-radius: 6px; }
.nav-item { padding: 10px; cursor: pointer; margin-bottom: 5px; }
.nav-item.active { background: #eef2ff; color: var(--primary); }
=====

```

- didalam folder views buat file **index.ejs** isi kode dibawah ini.

```

=====
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Admin System</title>
  <link rel="stylesheet" href="/styles.css">
</head>
<body>
  <aside>
    <h3>Admin Panel</h3>
    <div class="nav-item active" onclick="switchView('admin-view',
this)">Purchases</div>
    <div class="nav-item" onclick="switchView('chat-view', this)">AI Chat</div>
  </aside>

  <main>
    <!-- Purchase View -->
    <section id="admin-view" class="section active">
      <div class="grid">
        <div class="card">

```



```

        <h3>New Purchase</h3>
        <form id="purchaseForm">
            <input type="text" id="customerName" placeholder="Customer Name"
required>
            <select id="productSelect" required><option
value="">Loading...</option></select>
            <div id="stockLabel" style="margin-bottom:10px; font-size:0.9rem;
color:#666;"></div>
            <input type="number" id="quantity" min="1" value="1" required>
            <button>Buy Now</button>
        </form>
    </div>
    <div class="card">
        <h3>Stats</h3>
        <p>Active Orders: <b id="stat-orders">0</b></p>
        <p>Revenue: <b id="stat-revenue">$0.00</b></p>
    </div>
</div>
<div class="card">
    <h3>Transactions</h3>
    <table>

<thead><tr><th>ID</th><th>Customer</th><th>Product</th><th>Status</th><th>A
ction</th></tr></thead>
    <tbody id="purchaseTable"></tbody>
</table>
</div>
</section>

<!-- Chat View -->
<section id="chat-view" class="section">
    <div class="card">
        <h3>AI Assistant</h3>
        <div class="chat-container">
            <div class="chat-messages" id="chatMsgs">
                <div class="message msg-ai">Hello! I can answer admin
questions.</div>
            </div>
            <div style="padding:10px; border-top:1px solid #eee; display:flex;
gap:10px;">
                <input type="text" id="chatInput" placeholder="Ask AI..."
onkeypress="if(event.key==='Enter') sendChat()">
                <button onclick="sendChat()">Send</button>
            </div>

```

```

        </div>
    </div>
</section>
</main>

<script>
    let productsCache = [];

    async function load() {
        // Load Products
        const res = await fetch('/api/products');
        productsCache = await res.json();
        const sel = document.getElementById('productSelect');
        sel.innerHTML = '<option value="">Choose Product</option>';
        productsCache.forEach(p => {
            const opt = document.createElement('option');
            opt.value = p.id; opt.textContent = `${p.name} (${p.price})`;
            sel.appendChild(opt);
        });
        refreshStats();
        refreshTable();
    }

    async function refreshStats() {
        const r = await fetch('/api/stats');
        const d = await r.json();
        document.getElementById('stat-orders').textContent = d.activeOrders;
        document.getElementById('stat-revenue').textContent =
`${d.revenue.toFixed(2)}`;
    }

    async function refreshTable() {
        const r = await fetch('/api/purchases');
        const d = await r.json();
        const tb = document.getElementById('purchaseTable');
        tb.innerHTML = "";
        d.forEach(p => {
            const btn = p.status === 'Active' ? `<button class="btn-danger"
onclick="cancel(${p.id})">Cancel</button>` : '-';
            tb.innerHTML +=
`<tr><td>${p.id}</td><td>${p.customer}</td><td>${p.productName}</td><td>${p.st
atus}</td><td>${btn}</td></tr>`;
        });
    }

```

```

document.getElementById('purchaseForm').onsubmit = async (e) => {
  e.preventDefault();
  const data = {
    customer: document.getElementById('customerName').value,
    productId: parseInt(document.getElementById('productSelect').value),
    quantity: parseInt(document.getElementById('quantity').value)
  };
  const r = await fetch('/api/purchase', {
    method: 'POST', headers: {'Content-Type': 'application/json'}, body:
JSON.stringify(data)
  });
  if(r.ok) { alert('Success!'); load(); }
  else { const x = await r.json(); alert(x.error); }
};

async function cancel(id) {
  if(!confirm('Cancel order?')) return;
  await fetch(`/api/cancel/${id}`, {method: 'POST'});
  load();
}

async function sendChat() {
  const inp = document.getElementById('chatInput');
  const txt = inp.value; if(!txt) return;
  document.getElementById('chatMsgs').innerHTML += `<div class="message
msg-user">${txt}</div>`;
  inp.value = "";
  const r = await fetch('/api/chat', {
    method: 'POST', headers: {'Content-Type': 'application/json'}, body:
JSON.stringify({message: txt})
  });
  const d = await r.json();
  document.getElementById('chatMsgs').innerHTML += `<div class="message
msg-ai">${d.reply}</div>`;
  document.getElementById('chatMsgs').scrollTop = 9999;
}

function switchView(id, nav) {
  document.querySelectorAll('.section').forEach(e=>e.classList.remove('active'));
  document.getElementById(id).classList.add('active');
  document.querySelectorAll('.nav-
item').forEach(e=>e.classList.remove('active'));

```

```

        if(nav) nav.classList.add('active');
    }

    // Update stock label
    document.getElementById('productSelect').onchange = function() {
        const p = productsCache.find(x => x.id == this.value);
        document.getElementById('stockLabel').textContent = p ? `Stock: ${p.stock}` :
";
    };

    load();
</script>
</body>
</html>

```

=====

B. RUN application (UNTUK TEST)

Pada command prompt
node server.js

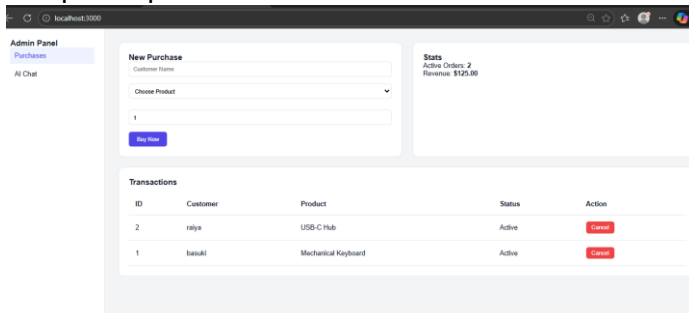
api tidak jalan karna saya tidak punya quota open ai

```

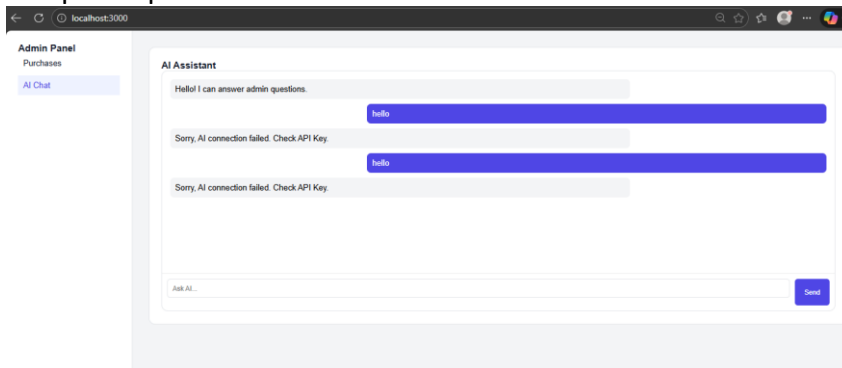
PS C:\admin-system> node server.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: https://dotenvx.com/ops
Server running at http://localhost:3000
I Error: {
  error: {
    message: 'You exceeded your current quota, please check your plan and billing details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.',
    type: 'insufficient_quota',
    param: null,
    code: 'insufficient_quota'
  }
}
I Error: {
  error: {
    message: 'You exceeded your current quota, please check your plan and billing details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.',
    type: 'insufficient_quota',
    param: null,
    code: 'insufficient_quota'
  }
}

```

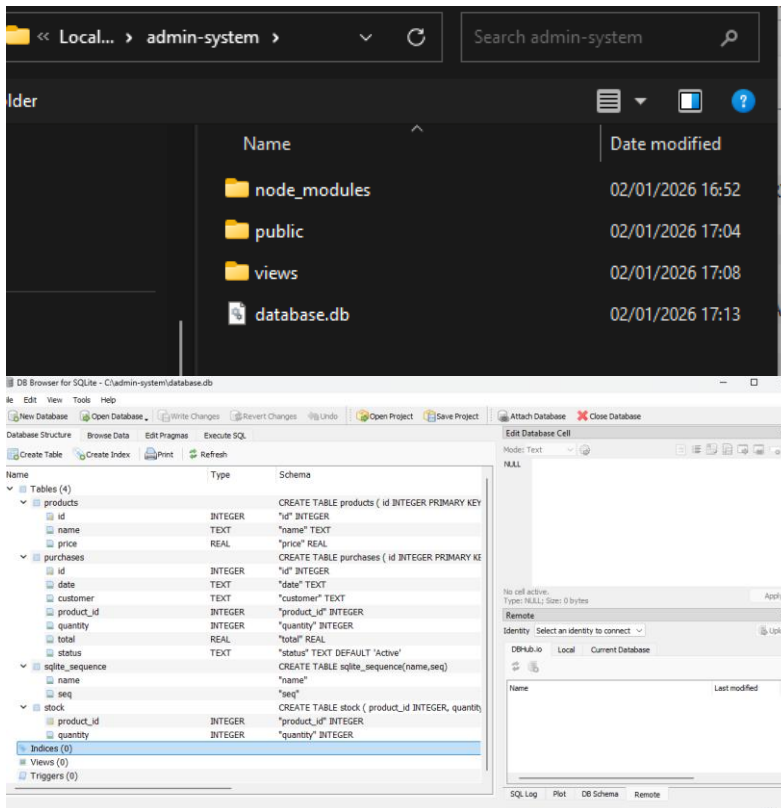
Tampilan aplikasi



Tampilan aplikasi ai chat

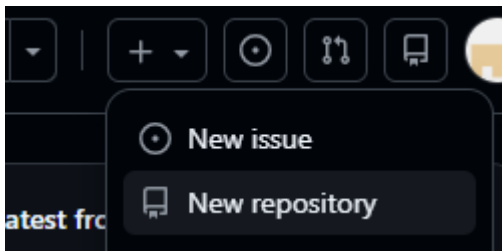


C. Struktur Database



D. Upload GitHub

- Alamat Github yang sudah jadi bisa di klik di <https://github.com/mbasuki/admin-system.git>
- klik New repository, klik create repository



Required fields are marked with an asterisk (*)

1 General

Owner * / Repository name *

admin-system is available.

Great repository names are short and memorable. How about [friendly-giggle?](#)

Description

67 / 350 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

Add README ☐ Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

Add license

Licenses explain how others can use your code. [About licenses](#)

Pada command prompt ketik dibawah ini

```
git init
```

```
git add .
```

```
git config --global user.name "mbasuki"
```

```
git config --global user.email ibaskorn@gmail.com
```

```
git commit -m "Initial upload of admin system"
```

```
git remote add origin https://github.com/mbasuki/admin-system.git
```

```
git branch -M main
```

```
git push -u origin main
```

```
PS C:\admin-system> git push origin main
```

```
Enumerating objects: 2478, done.
```

```
Counting objects: 100% (2478/2478), done.
```

```
Delta compression using up to 4 threads
```

```
Compressing objects: 100% (2330/2330), done.
```

```
Writing objects: 100% (2478/2478), 6.89 MiB | 509.00 KiB/s, done.
```

```
Total 2478 (delta 334), reused 0 (delta 0), pack-reused 0 (from 0)
```

```
remote: Resolving deltas: 100% (334/334), done.
```

```
To https://github.com/mbasuki/admin-system.git
```

* [new branch] main -> main
PS C:\admin-system>

E. Rekaman Aplikasi

<https://youtu.be/wBgrMglD0Ug>

F. Catatan

* Chat menggunakan open ai atau deepseek harus ada saldo, berikut hasil testnya

```
PS C:\admin-system> node server.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: ☒ audit secrets and track compliance: https://dotenvx.com/ops
Server running at http://localhost:3000
I Error: {
  error: {
    message: 'You exceeded your current quota, please check your plan and billing details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.',
    type: 'insufficient_quota',
    param: null,
    code: 'insufficient_quota'
  }
}
I Error: {
  error: {
    message: 'You exceeded your current quota, please check your plan and billing details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.',
    type: 'insufficient_quota',
    param: null,
    code: 'insufficient_quota'
  }
}
```

* nama file .env tidak bisa ditolak masuk ke GitHub karna berisi kode yang tidak dipublikasikan