

LAPORAN FINAL PROJECT
PENGEMBANGAN SISTEM OTOMATIS BERBASIS AI UNTUK ANALISIS
DAN PEMETAAN LAPORAN KEBERLANJUTAN PERUSAHAAN
ALGORITMA PEMROGRAMAN II



Dosen Pengajar:
Dr. Aziz Fajar, S.Kom., M.Kom.

Kelas SD-A1

Kelompok 12

Anggota Kelompok:

- | | |
|---|----------------|
| 1. Khairunnisa Keisha Anjani | NIM. 164241015 |
| 2. Muhammad Firdaus | NIM. 164241030 |
| 3. Mikael Ardiyanta Widyadana Purniawan | NIM. 164241031 |
| 4. Muhammad Iqbal Aulia Fattah | NIM. 164241052 |

UNIVERSITAS AIRLANGGA
FAKULTAS TEKNOLOGI MAJU DAN MULTIDISIPLIN
GANJIL 2025/2026

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR TABEL.....	3
BAB I	
PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Rumusan Masalah Proyek.....	6
1.3 Tujuan Proyek.....	7
BAB II	
TINJAUAN PUSTAKA.....	8
2.1 Landasan Teori.....	8
2.1.1 Konsep Laporan Keberlanjutan dan Standar Global.....	8
2.2 Kajian Proyek Sejenis.....	9
2.2.1 Pemanfaatan AI untuk Audit dan Otomatisasi Kepatuhan Pelaporan.....	9
2.2.2 Implementasi Integrasi Sistem untuk Efisiensi Alur Kerja.....	10
BAB III	
METODOLOGI PROYEK.....	11
3.1 Metodologi Pengembangan.....	11
3.3.2 Integrasi dan Alur Kerja.....	15
Pendekatan ini menjadikan sistem mampu bekerja tanpa keterlibatan manual, cepat, dan mudah direplikasi untuk banyak perusahaan.....	16
BAB IV	
HASIL DAN PEMBAHASAN.....	18
4.1 Implementasi Frontend.....	18
4.1.1 Arsitektur Frontend.....	18
4.1.2 Tampilan dan Fungsi Input.....	18
4.2 Implementasi Backend dan Pemrosesan Dokumen.....	19
4.2.1 Struktur Kode dan Inisialisasi.....	19
4.2.2 Fungsi Kunci def analyze().....	19
4.3 Implementasi Orkestrasi (n8n Workflow).....	23
4.4 Hasil Analisis dan Struktur Keluaran.....	24
4.4.1 Struktur Data Keluaran.....	24
BAB V	
KESIMPULAN DAN SARAN.....	25
5.1 Kesimpulan.....	25
5.2 Saran.....	25
DAFTAR PUSTAKA.....	27

DAFTAR TABEL

Tabel 3.1 Bahasa Pemrograman dan Framework.....	15
Tabel 3.2 Jadwal Rencana Pelaksanaan Pengembangan Sistem.....	17

DAFTAR GAMBAR

Gambar 3.1 Flowchart Desain Proses Sistem.....	13
Gambar 4.1 Kode frontend (HTML & CSS).....	18
Gambar 4.2 Tampilan dan Fungsi Input.....	19
Gambar 4.3 Struktur Kode dan Inisialisasi.....	19
Gambar 4.4 Kode Backend (app.py).....	20
Gambar 4.5 Kode Konfigurasi Entry Point dan Akses Jaringan Server.....	20
Gambar 4.6 File SR dari laptop 2 diterima oleh laptop 1 dan disimpan dalam folder /uploads.....	21
Gambar 4.7 Submisi dari Laptop 2 (klien).....	22
Gambar 4.8 Workflow n8n.....	23
Gambar 4.9 Hasil Analisis di Website.....	24

BAB I

PENDAHULUAN

1.1 Latar Belakang

Laporan keberlanjutan (*Sustainability Report*, SR) kini menjadi praktik umum bagi perusahaan besar sebagai wujud transparansi non-finansial. Survei Klynveld Peat Marwick Goerdeler (KPMG) 2024 menunjukkan sekitar 9 dari 10 perusahaan besar di dunia yang melaporkan kinerja keberlanjutan menggunakan standar *Global Reporting Initiative* (GRI) (GRI, 2024). Pada kawasan Asia-Pasifik, 81% perusahaan pun menerapkan standar GRI dalam laporannya (PWC, 2023). Pada sisi lain, standar *Sustainability Accounting Standards Board* (SASB) kini juga semakin vital karena fokus spesifiknya pada identifikasi isu-isu lingkungan, sosial, dan tata kelola (ESG) yang memiliki dampak finansial material bagi operasional perusahaan dan keputusan investasi (IFRS Foundation, 2023). Jika GRI menitikberatkan pada dampak perusahaan terhadap dunia, SASB melengkapinya dengan menyoroti dampak isu keberlanjutan terhadap kondisi keuangan perusahaan. Fakta ini mengindikasikan bahwa pengungkapan dampak ekonomi, lingkungan, dan sosial secara komprehensif telah menjadi bagian penting dari tata kelola perusahaan modern. Pelaporan keberlanjutan tidak hanya meningkatkan transparansi kepada pemangku kepentingan, tetapi juga membantu manajemen mengidentifikasi risiko dan peluang ESG di masa depan; di mana GRI memandu pelaporan materialitas dampak (GRI, 2024), sedangkan SASB menjembatani kinerja keberlanjutan dengan penciptaan nilai jangka panjang bagi investor.

Di sisi lain, proses menganalisis SR secara manual sangat menantang dan memakan waktu. SR disusun tanpa format baku dimana setiap perusahaan bebas menempatkan informasi sesuai tata letak dan gaya sendiri. Sebagai contoh, penelitian di Swedia mencatat bahwa laporan GRI tidak memiliki struktur baku sehingga pencarian data tertentu di dalamnya harus dilakukan secara manual untuk setiap laporan (Gutierrez-Bustamante & Espinosa-Leal, 2022). Akibatnya, konsultan atau analis harus membaca puluhan hingga ratusan halaman teks, tabel, dan grafik untuk memetakan informasi ke dalam kerangka standar keberlanjutan (misalnya GRI dan SASB). World Economic Forum menegaskan bahwa volume data keberlanjutan yang harus dilaporkan sangat besar dan kompleks, sehingga pekerjaan validasi dan pengungkapan informasi tersebut menjadi “sangat memakan sumber daya” jika dikerjakan secara manual (WEF, 2025). Analisis manual rentan bias dan memiliki perbedaan interpretasi antar analis, serta menghambat perusahaan dalam memenuhi tuntutan regulasi dan investor yang semakin ketat.

Di era ini, kemajuan teknologi *Artificial Intelligence* (AI) menawarkan peluang untuk mengatasi kendala tersebut. AI, terutama teknologi NLP (*Natural Language Processing*) dan *Machine Learning*, mampu mengotomatisasi banyak tugas analisis teks. Menggunakan AI, data dari berbagai laporan dapat digabungkan secara cepat, kesenjangan informasi dapat dideteksi, dan anomali dapat diidentifikasi tanpa intervensi manusia (WEF, 2025). Implementasinya, sistem AI dapat melakukan pencocokan teks otomatis antara isi laporan dengan indikator GRI atau SASB, memperkirakan kesesuaian “*fit*” materi pelaporan, serta menghasilkan ringkasan rekomendasi per topik material. Studi terbaru menunjukkan bahwa penggunaan *Large Language Model* (LLM) dapat secara signifikan mengurangi waktu dan tenaga yang dibutuhkan untuk menilai laporan keberlanjutan. Hanya dengan mengotomasi tugas-tugas yang memakan waktu ini, konsultan dan tim keberlanjutan dapat lebih banyak memfokuskan sumberdaya pada penerapan solusi nyata untuk pengurangan dampak lingkungan dan sosial, bukan sekadar pengumpulan data rutin (Wu, Y., Hu, P., & Wang, D. D., 2025).

Mengingat pentingnya analisis yang cepat dan akurat, diperlukan pengembangan sistem otomatis berbasis AI untuk menganalisis dan memetakan SR. Proyek ini bertujuan merancang solusi siap pakai yang mengintegrasikan platform otomasi n8n, antarmuka web berbasis Flask, serta penyimpanan hasil di Google Sheets. Sistem ini diharapkan dapat menerima laporan dalam format PDF, kemudian secara otomatis mengenali konten relevan, memetakannya ke standar GRI dan SASB, menilai tingkat kecocokannya, dan menyusun rekomendasi per topik material, sehingga mempercepat proses audit keberlanjutan dan meningkatkan kualitas keluaran analisis.

1.2 Rumusan Masalah Proyek

1. Bagaimana mengotomatisasi proses ekstraksi dan analisis informasi dari laporan keberlanjutan perusahaan (format PDF) menggunakan teknologi AI?
2. Bagaimana memetakan konten laporan keberlanjutan secara akurat ke dalam kerangka GRI dan SASB melalui sistem otomasi
3. Bagaimana sistem AI dapat menilai kecocokan (compliance) laporan dengan kriteria GRI dan SASB secara otomatis?

1.3 Tujuan Proyek

1. Membangun sistem otomatis berbasis AI yang mampu mengolah laporan keberlanjutan dalam format PDF menjadi data terstruktur.
2. Mengimplementasikan modul pemetaan konten laporan ke standar GRI dan SASB secara otomatis dalam alur kerja n8n.
3. Mengembangkan algoritma penilaian kecocokan laporan terhadap kriteria GRI dan SASB menggunakan ML atau NLP.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Bab Tinjauan Pustaka menyajikan landasan teoritis terhadap penelitian dan proyek sejenis yang relevan dengan pengembangan sistem otomatisasi analisis *Sustainability Reporting* (SR) berbasis *Artificial Intelligence*. Landasan ini bertujuan untuk memberikan konteks keilmuan yang kuat mengenai pelaporan keberlanjutan, teknologi AI, serta praktik otomatisasi alur kerja.

2.1.1 Konsep Laporan Keberlanjutan dan Standar Global

Laporan Keberlanjutan atau *Sustainability Reporting* (SR) adalah laporan yang secara berkala disusun perusahaan untuk mengkomunikasikan kinerja ekonomi, sosial, dan lingkungan secara terintegrasi sebagai bentuk pertanggungjawaban terhadap para pemangku kepentingan dan lingkungan (Global Reporting Initiative, 2016, dikutip dalam Fadillah & Harahap, 2023, hlm. 43). *Sustainability Report* menjadi krusial sebagai alat transparansi dan memberikan gambaran keberlanjutan dan penciptaan nilai jangka panjang.

Dalam konteks global, terdapat beberapa standar yang berfungsi sebagai acuan utama bagi perusahaan untuk menyusun laporan mereka, yaitu:

1. Global Standard Initiative (GRI): GRI merupakan salah satu standar internasional yang diakui dalam menghasilkan laporan keberlanjutan (Ananda dkk., 2023, hlm. 533). GRI didasarkan pada konsep *Triple Bottom Line* yang mencakup dimensi sosial, ekonomi, dan lingkungan keberlanjutan (Ananda dkk., 2023, hlm. 533).
2. Sustainability Accounting Standard Board (SASB): SASB adalah salah satu organisasi nirlaba untuk membantu dunia usaha dan investor mengembangkan pemahaman yang sama tentang dampak finansial dari keberlanjutan (Pratiwi, 2024). SASB beroperasi dengan keyakinan bahwa mengintegrasikan faktor lingkungan, sosial, dan tata kelola (ESG) ke dalam pelaporan keuangan dapat memberikan pandangan yang lebih komprehensif mengenai kinerja, ketahanan, dan prospek masa depan perusahaan (Nguyen, 2024).

2.1.2 Kecerdasan Buatan (AI) dan Pemrosesan Bahasa Alami (NLP)

Proyek ini memanfaatkan Kecerdasan Buatan atau *Artificial Intelligence* (AI) sebagai mesin analitik utama. Secara spesifik, teknologi ini didukung oleh:

1. Large Language Model (LLM): LLM merupakan suatu jenis model pembelajaran mesin yang dibangun untuk memahami, menganalisis, dan menghasilkan teks dalam bahasa alami (Wiradinata & Mawardi, 2025, hlm 4). Pada sistem ini, LLM berperan dalam membaca dan memahami konteks dari dokumen PDF yang besar (SR), mengekstraksi narasi yang relevan, dan mengklasifikasikan status cakupan laporan berdasarkan standar yang telah ditetapkan
2. Natural Language Processing (NLP): NLP adalah cabang kecerdasan buatan yang berfokus pada interaksi komputer dan bahasa manusia, baik dalam bentuk teks maupun suara sehingga mesin dapat memahami, menerjemahkan, dan menghasilkan bahasa alami seperti yang digunakan manusia sehari-hari (Supriady dkk., 2024, hlm 43).

2.2 Kajian Proyek Sejenis

Kajian proyek sejenis menyoroti bagaimana teknologi AI dan otomatisasi telah diterapkan untuk mengatasi tantangan yang serupa di ranah kepatuhan (*compliance*) dan efisiensi alur kerja.

2.2.1 Pemanfaatan AI untuk Audit dan Otomatisasi Kepatuhan Pelaporan

Salah satu contoh penggunaan AI dalam proses audit adalah penelitian “Penggunaan Berbagai Artificial Intelligence pada Proses Audit”. Penelitian ini melakukan tinjauan sistematis literatur tentang implementasi kecerdasan buatan dalam audit keuangan. Hasil penelitian ini menunjukkan bahwa penggunaan AI dan *robotic process automation* dapat meningkatkan efisiensi, akurasi, dan keandalan proses audit serta mempercepat pemeriksaan dokumen perusahaan. AI juga memudahkan deteksi pernyataan yang tidak sesuai fakta, analisis data volume besar, dan *compliance* terhadap standar akuntansi.

Salah satu contoh yang lain adalah penelitian “Penerapan Audit Berbasis Artificial Intelligence di Indonesia” (UNM, 2023) menggunakan metode meta sintesis berbagai studi kualitatif terkait penerapan AI untuk audit di Indonesia. Hasil

penelitian ini menunjukkan bahwa AI membawa manfaat dalam efisiensi, ketepatan waktu, serta kualitas audit yang lebih baik pada pelaporan keuangan dan operasional.

2.2.2 Implementasi Integrasi Sistem untuk Efisiensi Alur Kerja

Adapun berbagai penelitian yang mengimplementasi integrasi sistem untuk efisiensi alur kerja, salah satunya adalah penelitian “Penerapan Sistem Informasi untuk Meningkatkan Efisiensi Operasional dan Pengambilan Keputusan di Perusahaan” (Arujisaputra, 2025). Penelitian ini menjelaskan bahwa integrasi sistem informasi, seperti ERP, di berbagai fungsi bisnis (produksi, jadwal, distribusi, dan layanan pelanggan) memberikan penghematan waktu, meningkatkan visibilitas operasi, dan mengoptimalkan penggunaan sumber daya. Studi kasus juga membahas bagaimana pemantauan *real-time* dan otomatisasi *workflow* menghasilkan efisiensi operasional tinggi.

BAB III

METODOLOGI PROYEK

3.1 Metodologi Pengembangan

Pengembangan sistem proyek ini menggunakan metodologi *Agile* dengan pendekatan Scrum. Metode ini dipilih karena memberikan fleksibilitas terhadap perubahan kebutuhan sistem serta memungkinkan pengembangan dilakukan secara bertahap melalui iterasi (*sprint*) yang terukur. Menurut Schwaber dan Sutherland (2020), Scrum merupakan kerangka kerja Agile yang bersifat iteratif, adaptif, dan *incremental*, sehingga sesuai untuk sistem yang membutuhkan pengujian berulang serta evaluasi berkala pada kebutuhan pengguna.

Tahapan utama dalam metodologi ini meliputi:

1. Perumusan Kebutuhan (*Product Backlog*)

Kebutuhan sistem dicatat dalam *backlog* dan diprioritaskan berdasarkan fitur utama yang diperlukan. Fadhil & Aziz (2021) menjelaskan bahwa pemetaan kebutuhan yang baik pada tahap awal pengembangan algoritma AI dapat meningkatkan akurasi dan ketepatan sistem di tahap akhir.

2. Perencanaan Sprint (*Sprint Planning*)

Fitur-fitur dalam *backlog* disusun berdasarkan prioritas dan kompleksitas, kemudian dibagi menjadi beberapa *sprint* dengan durasi kurang lebih dua minggu.

3. Pengembangan Sistem (*Sprint Development*)

Tim pengembang proyek ini membangun fitur secara bertahap, meliputi *parsing* dokumen PDF, pemrosesan teks menggunakan *Large Language Model* (LLM), penyusunan basis data standar GRI/SASB, serta integrasi alur otomatis menggunakan n8n.

4. Pengujian dan Evaluasi (*Testing and Review*)

Pengujian dilakukan dengan menggunakan beberapa laporan keberlanjutan perusahaan untuk memastikan ketepatan pemetaan, konsistensi logika, dan keakuratan rekomendasi yang dihasilkan oleh AI.

5. Implementasi dan Perbaikan (*Deployment and Enhancement*)

Setelah sistem teruji, pengembang proyek melakukan penyempurnaan

berdasarkan umpan balik konsultan dan selanjutnya mempersiapkan sistem untuk digunakan oleh pengguna.

Berbekal siklus iteratif tersebut, sistem dapat disempurnakan secara bertahap hingga mencapai hasil yang stabil dan dapat diandalkan.

3.2 Perancangan Sistem

3.2.1 Arsitektur Sistem

Sistem dirancang berbasis alur otomatis yang mengintegrasikan formulir pengguna, kemampuan pemrosesan dokumen, kecerdasan buatan, serta penyimpanan hasil analisis. Pengguna mengunggah laporan keberlanjutan melalui antarmuka web, kemudian sistem secara otomatis memproses dokumen menggunakan *workflow engine* n8n dan mengirimkannya ke model kecerdasan buatan untuk dianalisis. Hasil analisis kemudian disimpan secara terstruktur pada Google Sheets sehingga mudah ditinjau oleh konsultan. Teknologi automasi seperti n8n efektif untuk menghubungkan beberapa layanan melalui API tanpa intervensi manual (Díaz dkk., 2018).

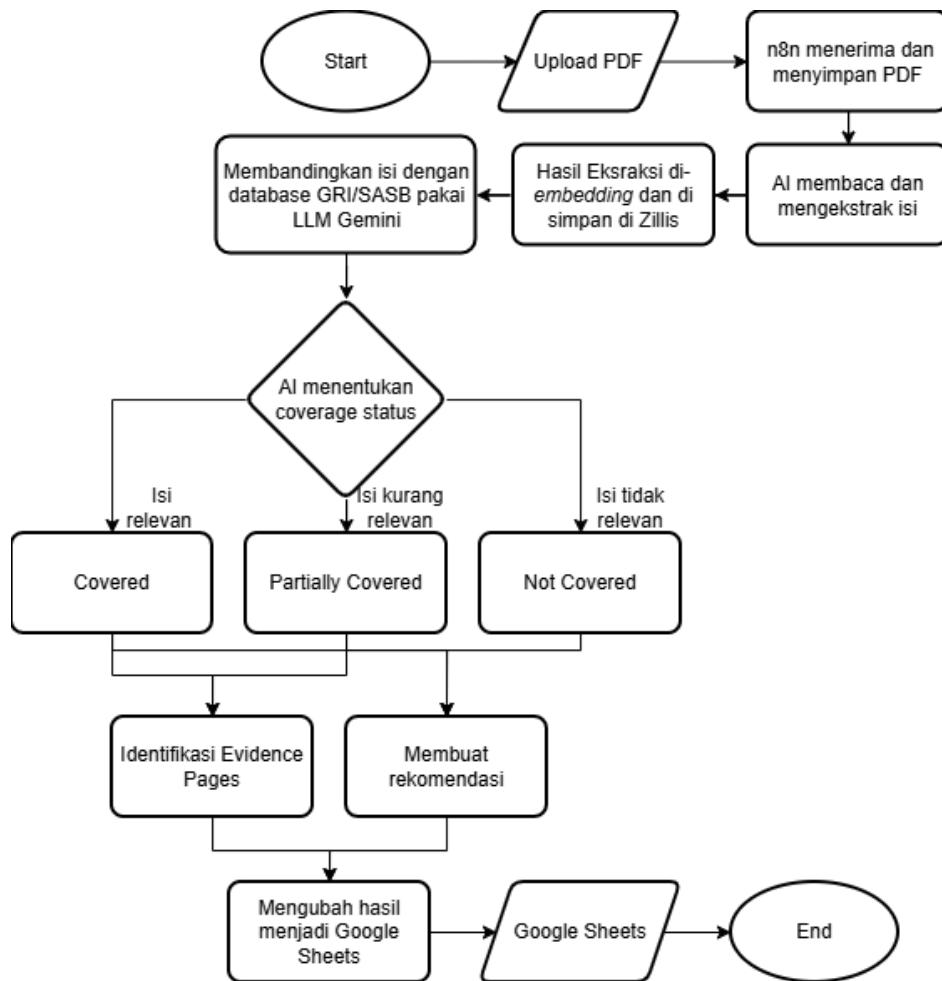
Alur arsitektur sistem secara garis besar terdiri atas:

1. Input laporan keberlanjutan oleh pengguna melalui formulir web
2. Penyimpanan otomatis dokumen ke Zillis
3. Pemanggilan model AI untuk proses pembacaan dokumen serta pemetaan ke standar GRI/SASB
4. Pembuatan ringkasan, status indikator, bukti halaman, dan rekomendasi
5. Penyimpanan hasil akhir ke Google Sheets secara otomatis

Arsitektur ini memungkinkan proses berjalan tanpa intervensi manual, sehingga mempercepat analisis dan mengurangi risiko kesalahan manusia (*human error*).

3.2.2 Desain Proses Sistem

Alur proses sistem dapat digambarkan melalui *flowchart* sebagai berikut:



Gambar 3.1 Flowchart Desain Proses Sistem

Berikut penjelasan alur sistem berdasarkan gambar tersebut:

1. PDF *Upload & Penyimpanan*

Proses dimulai ketika pengguna mengunggah file *Sustainability Report (SR)*. Kemudian n8n menerima *file* tersebut dan otomatis menyimpannya ke Zillis untuk diproses lebih lanjut.

2. *Document Processing*

AI kemudian membaca dan mengekstrak isi laporan, mencakup teks, tabel, maupun gambar. Tahap ini memastikan seluruh informasi dari laporan dapat dianalisis meskipun formatnya bervariasi (teks biasa atau hasil *scan*).

3. *Standards Analysis*

Konten yang sudah diekstrak dibandingkan dengan *database* standar GRI dan SASB sehingga memberikan label status sebagai berikut:

- a. Jika informasi sudah lengkap, status *Covered*.
- b. Jika sebagian terpenuhi, status *Partially Covered*.
- c. Jika tidak ditemukan data relevan, status *Not Covered*.

Jika seluruh atau sebagian informasi telah terpenuhi, AI juga mengidentifikasi halaman bukti dan menghasilkan rekomendasi untuk topik yang belum sesuai standar.

4. *Output Mapping*

Hasil analisis (status, bukti, rekomendasi) kemudian dipetakan ke format Google Sheets agar mudah dibaca dan digunakan oleh tim *consulting*. Sistem secara otomatis menyimpan hasil akhir tersebut ke *spreadsheet*.

3.2.3 Desain Basis Data

Sistem menggunakan basis data yang berisi daftar indikator GRI dan SASB beserta definisi, kode indikator, dan kata kunci pendukung. Basis data ini menjadi referensi utama dalam proses pemetaan isi laporan. GRI (2020) dan SASB (2021) menyatakan bahwa pelaporan keberlanjutan harus mencakup pengungkapan kualitatif dan kuantitatif yang dapat diverifikasi, sehingga sistem harus mampu mendeteksi keduanya.

Output akhir untuk setiap perusahaan disimpan dalam format tabel pada Google Sheets dengan komponen sebagai berikut:

1. Identitas perusahaan dan tahun laporan
2. Daftar *material topics*
3. Kode indikator GRI/SASB terkait
4. Status pemenuhan: *Covered*, *Partially Covered*, atau *Not Covered*
5. Halaman bukti dari laporan
6. Rekomendasi perbaikan atau kelengkapan data

Penyimpanan berbasis Google Sheets dipilih agar hasil analisis mudah dipantau, disunting, dan diintegrasikan dalam laporan konsultan.

3.3 Implementasi Sistem

3.3.1 Bahasa Pemrograman dan *Framework*

Implementasi sistem melibatkan beberapa teknologi sebagai berikut:

Komponen Sistem	Teknologi yang Digunakan
<i>Backend</i> (RESTful API)	Framework Flask (Python)
<i>Frontend</i>	HTML dan CSS
<i>Workflow</i> Otomatis	n8n
Pemrosesan Dokumen PDF	Python (pdfplumber / PyPDF)
Pemrosesan <i>Embedding</i>	Embedding Gemini
Penyimpanan Data (<i>Vector embeddings</i>)	Zillis dan Google Sheets API
Pemodelan AI	Gemini (<i>Large Language Model</i> berbasis Python)
Penyimpanan Data (Laporan Akhir)	Google Sheets

Tabel 3.1 Bahasa Pemrograman dan Framework

Penelitian oleh Wirohman dkk. (2021) menunjukkan bahwa automasi API dapat mempercepat proses integrasi data dan mengurangi kesalahan manual dalam pengolahan dokumen perusahaan.

3.3.2 Integrasi dan Alur Kerja

1. Pengguna mengunggah dokumen melalui formulir web
2. Sistem n8n menerima berkas dan memicu alur pemrosesan
3. Model AI melakukan ekstraksi teks dan analisis konten

4. Hasil teks yang telah diekstraksi akan di-*embedding* dan disimpan di Zillis
5. Sistem membandingkan konten laporan dengan indikator GRI/SASB pada basis data.
6. Keluaran berupa pemetaan indikator, status pemenuhan, dan rekomendasi dikirim ke Google Sheets
7. Konsultan dapat melihat hasil akhir pada format tabel secara *real-time*.

Pendekatan ini menjadikan sistem mampu bekerja tanpa keterlibatan manual, cepat, dan mudah direplikasi untuk banyak perusahaan.

3.4 Jadwal Pelaksanaan

Rencana pelaksanaan pengembangan sistem diproyeksikan dalam rentang waktu enam minggu dengan rincian sebagai berikut:

Minggu	Kegiatan Utama	Output/Target
1	<ul style="list-style-type: none"> - Analisis kebutuhan sistem - Pendefinisian masalah proyek - Penyusunan tujuan dan ruang lingkup - Studi literatur tentang GRI, SASB, AI NLP, dan <i>automation tools</i> - Penyusunan proposal produk/program 	<ul style="list-style-type: none"> - Draft proposal lengkap (Pendahuluan, Rumusan Masalah, Tujuan, Metodologi, Tinjauan Pustaka, Rancangan Solusi Awal)
2	<ul style="list-style-type: none"> - Perancangan sistem teknis - Pembuatan <i>flowchart</i> proses analisis - Perancangan arsitektur sistem (n8n → Zillis → AI → Sheets) - Penyusunan rancangan <i>database</i> standar GRI/SASB - Pembuatan <i>mock-up</i> UI form <i>upload & dashboard</i> hasil 	<ul style="list-style-type: none"> - Dokumen desain sistem lengkap - Flowchart proses otomatis - Diagram arsitektur sistem - <i>Mock-up</i> UI WebApp

3	<ul style="list-style-type: none"> - Implementasi dasar WebApp <i>upload PDF</i> - Integrasi ke Zillis & Sheets melalui n8n - Proses <i>parsing</i> dokumen dan pemanggilan AI awal - Pembuatan prototype <i>workflow</i> utama (<i>upload → analisis → spreadsheets</i>) 	<ul style="list-style-type: none"> - Prototype berjalan untuk beberapa laporan uji coba - Workflow AI dan penyimpanan data sudah berfungsi
4	<ul style="list-style-type: none"> - Penyempurnaan WebApp dan UI - Finalisasi proses analisis dan pemetaan indikator - Pengujian semua fitur dengan beberapa dokumen perusahaan - Deployment WebApp agar dapat diakses publik - Persiapan materi presentasi final dan demo <i>live</i> 	<ul style="list-style-type: none"> - WebApp live dan siap digunakan - Analisis otomatis berfungsi (<i>upload → AI → spreadsheets</i>) - Presentasi akhir dan demo siap

Tabel 3.2 Jadwal Rencana Pelaksanaan Pengembangan Sistem

Dengan jadwal tersebut, sistem diharapkan dapat diselesaikan tepat waktu dan siap digunakan untuk mendukung proses analisis keberlanjutan secara otomatis.

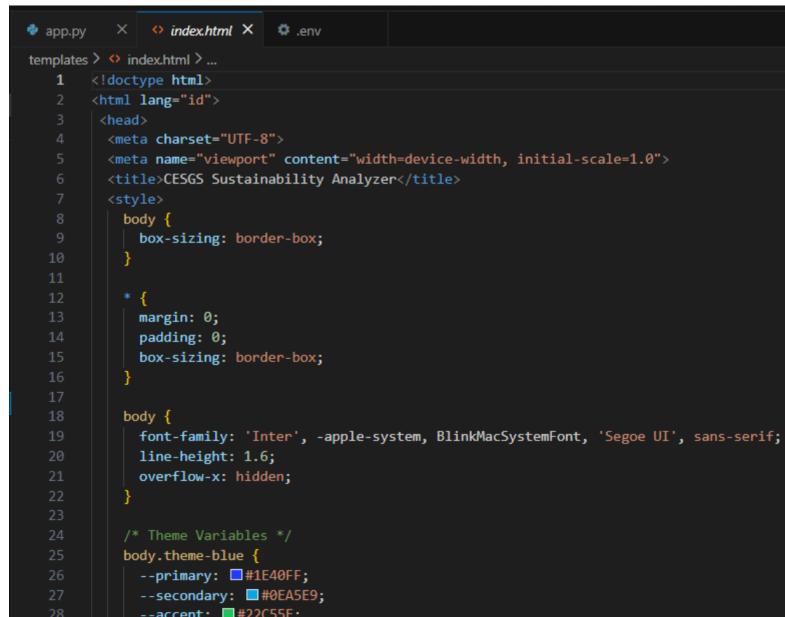
BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi *Frontend*

4.1.1 Arsitektur *Frontend*

Frontend sistem dikembangkan menggunakan HTML, CSS, dan disajikan oleh backend Flask. Arsitektur ini bertanggung jawab untuk menampilkan formulir input yang menerima data penting dari pengguna.

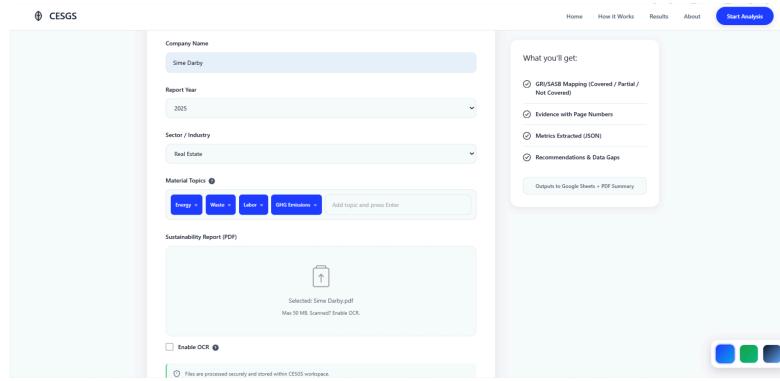


```
app.py      index.html      .env
templates > index.html > ...
1  <!doctype html>
2  <html lang="id">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>CESGS Sustainability Analyzer</title>
7  <style>
8  body {
9    box-sizing: border-box;
10   }
11  *
12   {
13     margin: 0;
14     padding: 0;
15     box-sizing: border-box;
16   }
17  body {
18    font-family: 'Inter', -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
19    line-height: 1.6;
20    overflow-x: hidden;
21  }
22  /* Theme Variables */
23  body.theme-blue {
24    --primary: #1E40FF;
25    --secondary: #0EA5E9;
26    --accent: #22C55E;
27  }
28
```

Gambar 4.1 Kode *frontend* (HTML & CSS)

4.1.2 Tampilan dan Fungsi Input

Frontend memungkinkan pengguna memasukkan detil laporan, yang merupakan data penting untuk memicu proses analisis, seperti nama perusahaan, tahun laporan diterbitkan, Sektor, dan *Material Topics*. Data ini kemudian dikirim melalui HTTP POST ke endpoint /analyze di *backend*.



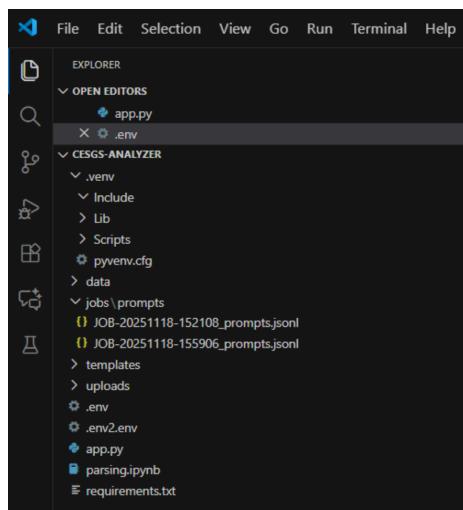
Gambar 4.2 Tampilan dan Fungsi Input

4.2 Implementasi *Backend* dan Pemrosesan Dokumen

Implementasi *backend* sistem ini dibangun menggunakan *framework* Flask berbasis Python. *Backend* berfungsi sebagai RESTful API yang menerima input dari antarmuka web (*frontend*) dan memulai proses analisis AI.

4.2.1 Struktur Kode dan Inisialisasi

Struktur *file* proyek diatur untuk mendukung alur kerja yang terstruktur, memisahkan logika aplikasi utama (app.py), penyimpanan data hasil *parsing* (data/chunks), dan *prompt* untuk model bahasa besar (jobs/prompts).



Gambar 4.3 Struktur Kode dan Inisialisasi

4.2.2 Fungsi Kunci def analyze()

Fungsi inti dalam app.py adalah def analyze(), yang dipicu oleh permintaan POST dari frontend. Fungsi ini mengorkestrasikan langkah-langkah pemrosesan dokumen.

```

app.py > ...
347 def analyze():
348     ...
349     "enable_ocr": enable_ocr,
350     "material_topics": material_topics,
351     "file_saved": bool(saved_path),
352     "file_path": str(saved_path) if saved_path else None,
353     "pdf_info": pdf_info,
354     "prompt_file_path": str(prompt_path), # untuk GPT Batch
355     "chunks_file_path": chunks_file_path, # untuk embedding & Zilliz
356     "mappings": mappings,
357 }
358
359 return jsonify(
360     {
361         "status": "success",
362         "job_id": job_id,
363         "message": "Job received. Prompt JSONL & chunks generated.",
364         "result": result,
365     }
366 )
367
368 except Exception as e:
369     return jsonify({"status": "error", "message": f"Server error: {e}"})
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454

```

Gambar 4.4 Kode *Backend* (app.py)

1. Penerimaan & Penyimpanan: Fungsi menerima *file* PDF dan parameter, kemudian menyimpan PDF di uploads/.
2. Pemrosesan Teks: File PDF diproses, diekstraksi teksnya, dan dipotong menjadi text chunks. Proses ini menggunakan pustaka Python seperti pdfplumber atau PyPDF.
3. Persiapan LLM: *Chunk* teks digunakan untuk membuat *file* JSONL Prompt. File ini berisi instruksi dan data yang siap dikirim secara batch ke API LLM untuk dianalisis.
4. Integrasi *Embedding* (RAG): *Embeddings* dari *chunk* teks juga dihitung menggunakan model Gemini dan disimpan ke Zilliz. *Vector database* ini memungkinkan sistem melakukan Retrieval-Augmented Generation (RAG) untuk mencari bukti halaman yang paling relevan dengan indikator GRI/SASB.

4.2.3 Arsitektur Jaringan dan Skenario Dua Perangkat (Client–Server)

```

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000, debug=True)

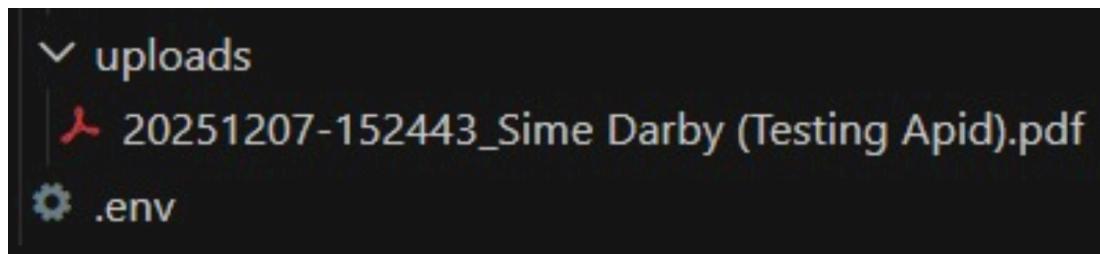
```

Gambar 4.5 Kode Konfigurasi Entry Point dan Akses Jaringan Server

Pada implementasi proyek ini menggunakan skenario dua perangkat yang saling terhubung dalam satu jaringan lokal (*local area network/LAN*) melalui Wi-Fi. Skenario ini dirancang agar sistem dapat diakses oleh pengguna dari perangkat lain, tetapi seluruh proses

komputasi tetap dijalankan pada laptop peneliti sebagai server. Secara garis besar, peran masing-masing perangkat adalah sebagai berikut:

1. Laptop 1 (Server / *Backend*)



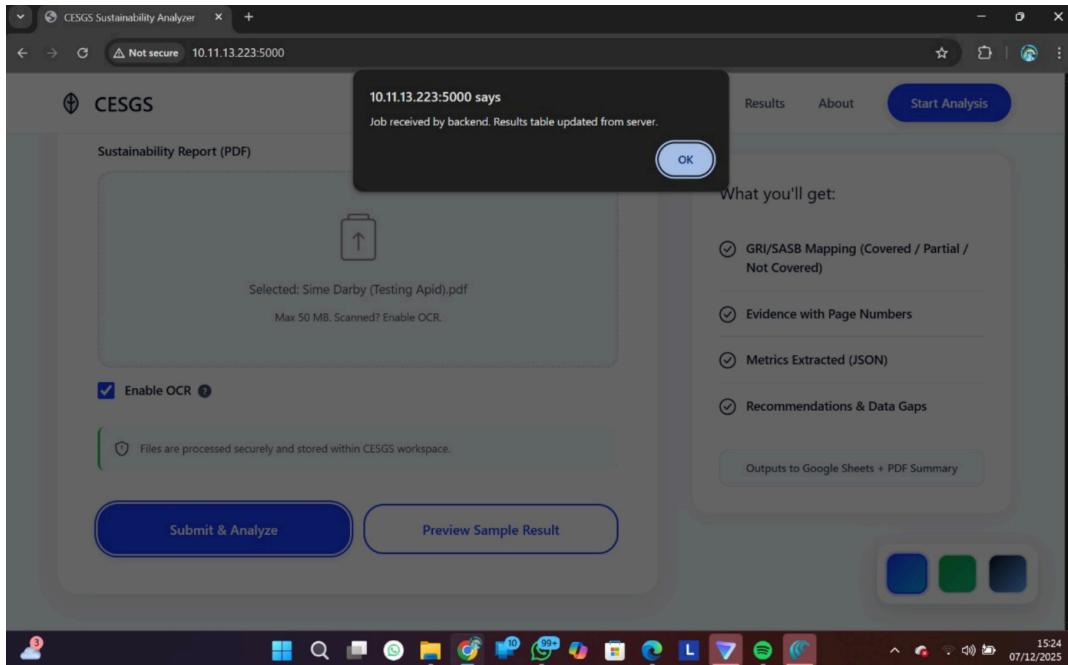
Gambar 4.6 File SR dari laptop 2 diterima oleh laptop 1 dan disimpan dalam folder /uploads

Menjalankan aplikasi backend berbasis Flask yang mengelola:

- a. Penerimaan input dari pengguna (formulir web dan unggah *file* PDF),
- b. Penyimpanan laporan keberlanjutan dalam folder tertentu,
- c. Pemrosesan awal (ekstraksi teks, *chunking*),
- d. Pemanggilan *worker AI* (*embedding* dengan Gemini dan penyimpanan vektor ke Zilliz),
- e. Penyusunan hasil analisis dalam format terstruktur (JSON).

Pada sisi Flask, server dikonfigurasi dengan perintah “app.run(host="0.0.0.0", port=5000, debug=True)”. Pengaturan host="0.0.0.0" membuat aplikasi dapat menerima permintaan (*request*) dari alamat IP lain di dalam jaringan Wi-Fi yang sama, bukan hanya dari *localhost*.

2. Laptop 2 / Perangkat Lain (*Client / Frontend*)



Gambar 4.7 Submisi dari Laptop 2 (klien)

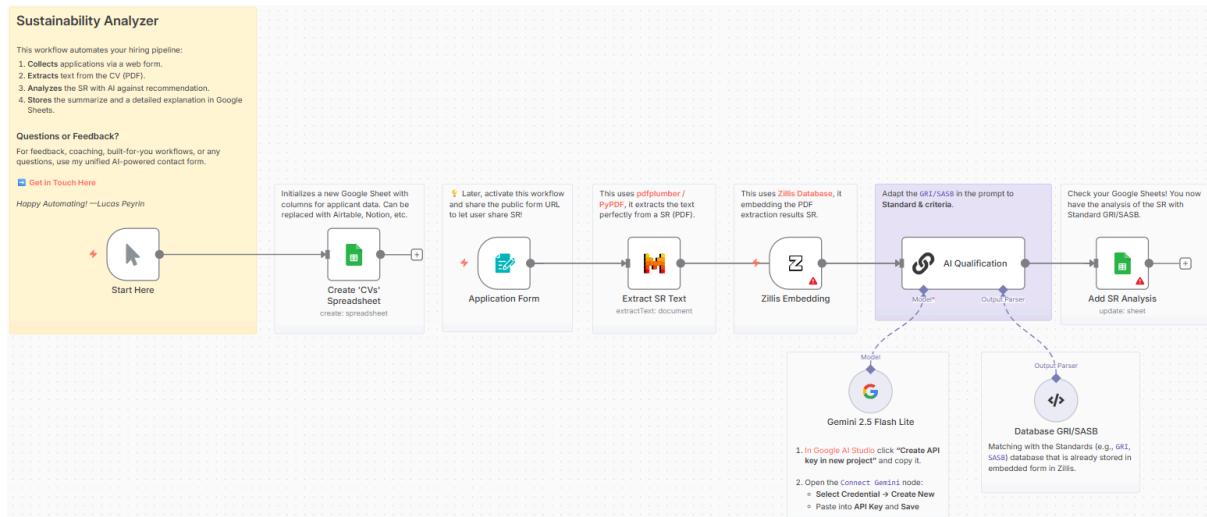
- a. Berperan sebagai klien yang hanya membutuhkan peramban web (*browser*) tanpa instalasi *software* tambahan.
- b. Pengguna mengakses sistem dengan mengetik alamat: `http://<IP_laptop_server>:5000` di mana `<IP_laptop_server>` adalah alamat IP lokal laptop server, misalnya `192.168.1.10:5000`. Alamat IP ini diperoleh melalui perintah seperti `ipconfig` (Windows) atau `ifconfig` (Linux/Mac).
- c. Seluruh input dilakukan melalui antarmuka web: pengguna mengisi nama perusahaan, tahun laporan, sektor, *material topics*, serta mengunggah *file PDF* laporan keberlanjutan.

Dengan konfigurasi tersebut, seluruh beban pemrosesan (ekstraksi PDF, *embedding*, *query* ke basis data vektor, dan analisis AI) tetap berjalan di laptop server, sedangkan perangkat klien hanya mengirim permintaan HTTP dan menerima hasil analisis dalam bentuk tampilan web. Pola ini mengikuti arsitektur client-server dan sekaligus mengimplementasikan konsep RESTful API, di mana klien mengirimkan permintaan POST ke endpoint `/analyze` pada server Flask, server memproses permintaan tersebut, dan server mengembalikan respons dalam bentuk data JSON yang kemudian ditampilkan kembali sebagai tabel dan ringkasan pada browser klien.

Skenario dua perangkat dalam satu jaringan Wi-Fi ini mensimulasikan kondisi sistem ketika diakses oleh banyak pengguna, tetapi tetap mempertahankan kontrol penuh atas data dan pemrosesan di sisi peneliti. Pendekatan ini juga memudahkan demonstrasi sistem di kelas atau di depan dosen penguji, karena dosen dapat mengakses aplikasi dari perangkatnya sendiri sementara seluruh logika dan komputasi tetap tersentralisasi pada laptop pengembang.

4.3 Implementasi Orkestrasi (n8n Workflow)

n8n digunakan sebagai *workflow engine* untuk mengotomatisasi alur kerja *end-to-end* tanpa intervensi manual. Alur kerja di n8n memastikan bahwa proses analisis berjalan secara sekuensial dan logis:



Gambar 4.8 *Workflow n8n*

1. Input: N8n menerima *file* dari *backend* dan menyimpannya di Zillis
2. Pemrosesan Dokumen: Model AI membaca dan mengekstrak SR pdf (termasuk teks, tabel, dan gambar).
3. *Embedding*: Hasil Ekstraksi akan di-*embedding* dan disimpan di Zillis
4. *Standards Analysis*: Proses inti ini terjadi di mana LLM Gemini membandingkan konten laporan dengan Basis Data Standar GRI/SASB, lalu LLM menentukan *Coverage Status*.
5. Keluaran Terstruktur: Berdasarkan status, sistem:
 - a. Mengidentifikasi Halaman Bukti (*Evidence Pages*).
 - b. Menghasilkan Rekomendasi Perbaikan (*Generate Recommendations*).
 - c. Melakukan *Mapping Results* dan menyimpan hasil ke Google Sheets.

4.4 Hasil Analisis dan Struktur Keluaran

Hasil akhir dari proses otomatis ini disajikan secara rapi dalam format Google Sheets. Platform ini dipilih karena memudahkan pemantauan dan integrasi data.

4.4.1 Struktur Data Keluaran

Tabel hasil analisis menyediakan data terstruktur yang sebelumnya harus diisi manual.

1. Kode Indikator: Daftar kode GRI/SASB yang relevan.
2. Status Pemenuhan: Hasil penilaian otomatis LLM: *Covered*, *Partially Covered*, atau *Not Covered*.
3. Halaman Bukti: Nomor halaman spesifik di PDF tempat bukti pemenuhan ditemukan.
4. Rekomendasi Perbaikan: Saran otomatis dari AI untuk menutup kesenjangan data (*data gaps*) atau meningkatkan kepatuhan.

The screenshot shows the CESGS platform interface. At the top, there's a navigation bar with links to Home, How it Works, Results, About, and a prominent blue 'Start Analysis' button. Below the navigation is a 'Status Panel' showing progress in Parsing, Chunking, Retrieval, Mapping, and Writing to Sheets, with a job ID: JOB-20281119-133002. The main content area is divided into sections: 'Report Summary' (File: Sime Darby.pdf, Pages: 135, First page excerpt: SUSTAINABILITY REPORT2023), 'Results Snapshot' (Material Topic: Energy, Framework: GRI, Code: 302-1, Status: Partial, Pages: 18-21, Recommendation: Report renewable vs non-renewable; add 302-3 energy intensity target.), and another row of results for Waste, Labor, and GHG Emissions. At the bottom right are three colored circular icons.

Gambar 4.9 Hasil Analisis di Website

	B	C	D	E	F	G	H
1							
2	Raters Score		-	18.4 (Low Risk)	18.3 (Low Risk)	21.2 (Medium Risk)	14.8 (Low Risk)
3	Latest SR		Tata Realty.pdf	Sime Darby.pdf	Longfor Holdings.pdf	Lippo Karawaci.pdf	BSD City.pdf
4	ESG Pillars	GRI	Tata Realty	Sime Darby Property Bhd.	Longfor Group Holdings Ltd.	PT Lippo Karawaci Tbk	PT Bumi Serpong Damai Tbk
5	Environmental	Energy	Emissions and Energy Management	Energy and Carbon Management	Energy Conservation and Emission Reduction	Energy Efficiency & Emissions	
6		Emissions	Emissions and Energy Management	Energy and Carbon Management	Energy Conservation and Emission Reduction	Energy Efficiency & Emissions	
7		Water	Waste Management	Circularity	Waste Management and Pollution Prevention	Waste Management	Circularity
8		Water and Effluents	Water Management	Water Management	Water Resource Management	Water Stewardship	Water Management
9		Biodiversity	Biodiversity	Urban Biodiversity	Biodiversity		
10		Supplier Environmental Assessment					
11							
12			Climate Change Adaption	Climate Adaption	Response to Climate Change		Climate Adaption
13					Design and Application of Green Building		Green Building
14			Sustainable Buildings				
15	Social	Employment					
16		Labor/Management Relations					
17		Occupational Health and Safety	Health and Safety	Occupational Health and Safety	Occupational Safety and Health	Occupational Health and Safety	Occupational Health and Safety
18		Training and Education	Employee Development	Talent Management and Training Development	Employee Career Development	Training and Development	Training and Education
19		Diversity and Equal Opportunity	Diversity and Inclusion	Diversity and Inclusion			
20		Non-discrimination		Labour Standards and Practices	Protection of Employees Rights and Interest	Employment & Rights	
21		Freedom of Association and Collective Bargaining					

Gambar 4.10 Output Tabel Google Sheet masing-masing SR Perusahaan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian proyek, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Sistem berhasil mengotomatisasi proses ekstraksi informasi dari laporan keberlanjutan berformat PDF. Integrasi antara antarmuka web berbasis HTML dan CSS serta *backend* berbasis Flask (Python) sebagai Restful API dengan *workflow engine* n8n memungkinkan sistem menerima dokumen, melakukan pemotongan teks (*chunking*), dan menyimpannya ke dalam basis data vektor (Zilliz) tanpa intervensi manual yang signifikan.
2. Pemanfaatan *Large Language Model* (LLM) Gemini yang didukung oleh teknik *Retrieval-Augmented Generation* (RAG) terbukti mampu memetakan konten laporan yang tidak terstruktur ke dalam kerangka kerja baku GRI dan SASB. Sistem mampu mengenali konteks narasi perusahaan dan mencocokkannya dengan indikator spesifik pada basis data standar yang telah disiapkan.
3. Sistem mampu menilai tingkat kecocokan (*compliance*) laporan dengan memberikan status terukur, yaitu *Covered*, *Partially Covered*, atau *Not Covered*. Selain itu, sistem berhasil menghasilkan keluaran yang bernilai tambah berupa identifikasi nomor halaman bukti (*evidence pages*) dan penyusunan rekomendasi perbaikan untuk menutup kesenjangan data (*data gaps*).
4. Output akhir sistem yang terintegrasi langsung ke Google Sheets memberikan format yang terstruktur dan mudah diakses. Hal ini menjawab permasalahan efisiensi waktu bagi konsultan atau auditor, karena data yang sebelumnya harus dipetakan secara manual dari ratusan halaman kini tersedia dalam bentuk tabel siap pakai secara *real-time*.

5.2 Saran

Pengembangan sistem ke depan disarankan fokus pada peningkatan analisis data visual (infografis) dan penambahan standar global baru selain GRI/SASB. Selain itu, integrasi *dashboard* web interaktif serta fitur validasi manual (*human-in-the-loop*) sangat

diperlukan untuk menjamin akurasi interpretasi AI dan memberikan pengalaman pengguna yang lebih baik.

DAFTAR PUSTAKA

- Ananda, W., Pradesa, H. A., & Wijayanti, R. (2023). Pelaksanaan Sustainability Report Berdasarkan GRI Standards Guidelines Pada Perusahaan Manufaktur di Indonesia. *Ekonomi, Keuangan, Investasi dan Syariah (EKUITAS)*, 5(2), 533.
- Arujisaputra, E. T. (2025). Penerapan Sistem Informasi untuk Meningkatkan Efisiensi Operasional dan Pengambilan Keputusan di Perusahaan. *Journal Scientific of Mandalika*, 6(3), 700-709.
- Bazin, S., & Hayes, M. (2025, September 26). *How AI can transform sustainability reporting* | World Economic Forum. The World Economic Forum. Retrieved November 11, 2025, from <https://www.weforum.org/stories/2025/09/harnessing-ai-for-sustainability-reporting-path-forward/>
- Bustamante, M. G., & Leal, L. E. (2022). Natural language processing methods for scoring sustainability reports—A study of Nordic listed companies. *Sustainability* 2022, 14(15), 1–26. <https://doi.org/10.3390/su14159165>
- Díaz, J., Alvarez, L., & Torres, M. (2018). Workflow automation for document processing in enterprise systems. *Journal of Information Systems*.
- Fadhil, M., & Aziz, A. (2021). Designing AI-assisted document classification systems. *Journal of Computational Intelligence*.
- Fadillah, M. A., & Harahap, L. (2023). Sustainability reporting di Indonesia: Kebermanfaatannya bagi stakeholders dan shareholders (Studi empiris pada PT Bank Central Asia Tbk Periode 2021 — 2022). *Proceeding Auditing and Accounting Conference*, 42.
- GRI. (2020). *Global Reporting Initiative Standards*. GRI Publications.
- GRI. (2024, November 27). *GRI global adoption by top companies continues to grow*. Retrieved November 11, 2025, from <https://www.globalreporting.org/news/news-center/gri-global-adoption-by-top-companies-continues-to-grow/>
- IFRS Foundation. (2023). *SASB Standards: Connecting Sustainability to Financial Performance*. International Financial Reporting Standards Foundation. Diakses dari <https://sasb.org/>
- Indonesia - ESG Team, P. (n.d.). *Sustainability Counts II*.

Nguyen, A. (2024, April 26). *Understanding the SASB: Sustainability Accounting Standards Board*. Seneca ESG. Retrieved November 9, 2025, from <https://senecaesg.com/id/insights/understanding-the-sasb-sustainability-accounting-standards-board/>

Pratiwi, I. (2024, March 6). *Standar umum dan kerangka kerja sustainability report*. Pratama Institute. Retrieved November 9, 2025, from <https://pratamainstitute.com/standar-umum-dan-kerangka-kerja-sustainability-report>

SASB. (2021). *Sustainability Accounting Standards Board framework*. SASB Publications.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.

Wiradinata, A. S., & Mawardi, V. C. (2025). Abstractive text summarization berita bahasa Indonesia menggunakan Retrieval-Augmented Generation. *Jurnal Ilmu Komputer dan Sistem Informasi*, 4(1).

Wirohman, A., Sutopo, D., & Ramadhan, H. (2021). API-based automation for corporate data integration. *International Journal of Information Technology*.

Wu, Y., Hu, P., & Wang, D. D. (2025). The AI annotator: Large language models' potential in scoring sustainability reports. *Systems* 2025, 13(10), 1–28. <https://www.google.com/search?q=https://doi.org/10.3390/systems13100899>