



Task Ly

II3160 - Teknologi Sistem Terintegrasi

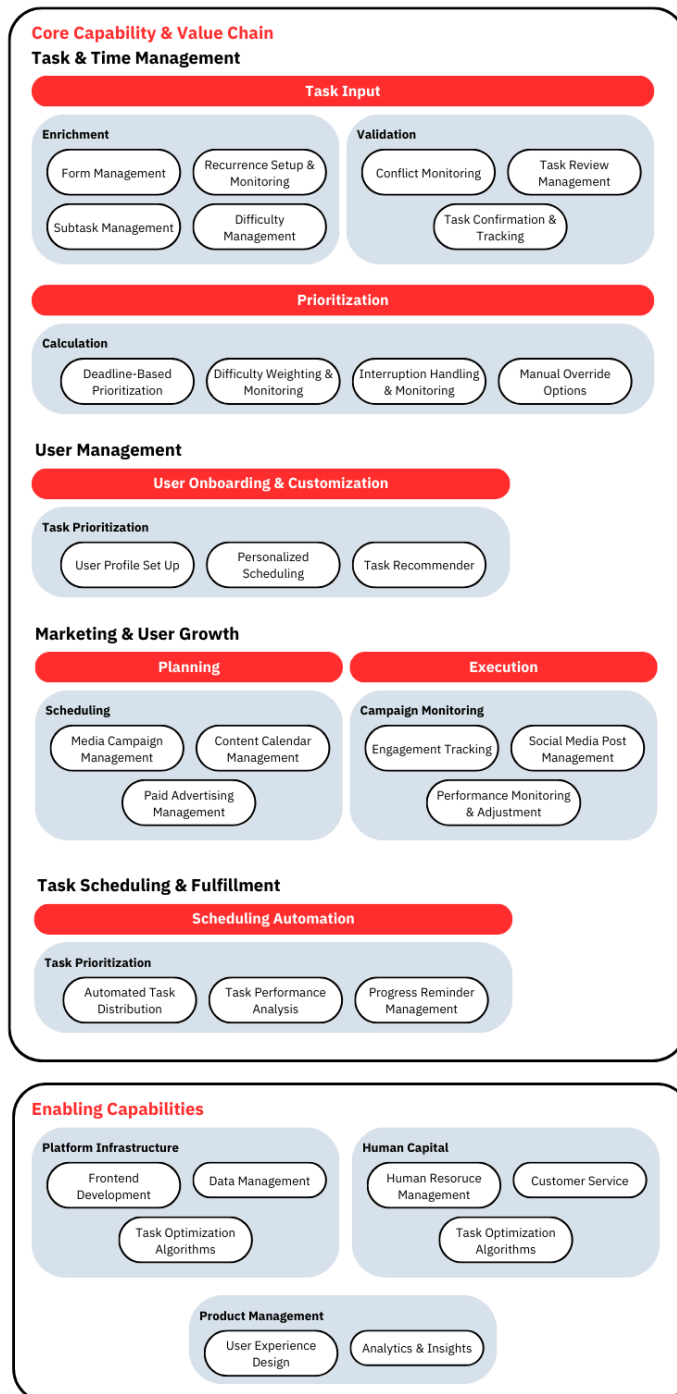
Pengembangan Sistem yang mampu
menyederhanakan aktivitas harian
dan **mengelola waktu** dengan lebih
baik.

Jihan Aurelia
18222001

DAFTAR ISI

DAFTAR ISI.....	2
A. Business Capability Map.....	3
B. Sub-domain.....	4
C. Software Architecture (Tech Stack).....	4
1. Frontend.....	4
2. Backend.....	5
3. Penyimpanan Data.....	6
4. Algoritma Penjadwalan.....	7
5. Visualisasi.....	8
6. Hosting dan Deployment.....	8
D. Langkah Implementasi.....	9
E. Fitur.....	11
F. Certificate HTTPS.....	12
G. Integrasi Service.....	12
1. Menggunakan Service.....	12
2. Menyediakan Service.....	14
H. Kesimpulan.....	15

A. Business Capability Map



Tujuan Bisnis:

Membantu pengguna mengatur dan mengelola tugas mereka secara efisien dengan penjadwalan otomatis berdasarkan input seperti jenis tugas, beban kerja, deadline, dan tingkat kesulitan, tanpa memerlukan layanan backend.

Core Unique Capabilities:

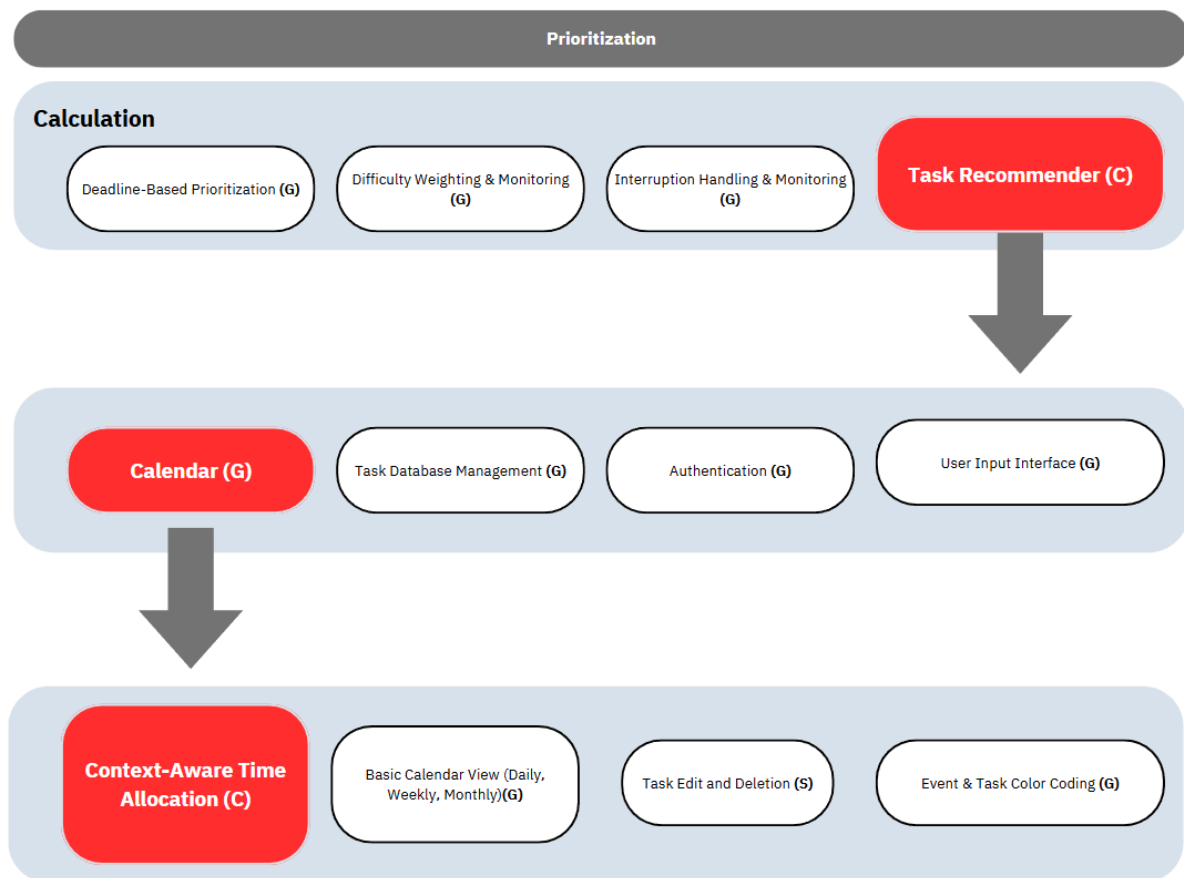
Otomatis mengatur dan memberikan rekomendasi jadwal mingguan berdasarkan masukan user, yaitu

- Nama tugas
- Jumlah *deliverables*
- Deadline
- Prioritas
- Tingkat kesulitan

Fitur Utama:

Task Recommender

B. Sub-domain



C. Software Architecture (Tech Stack)

1. Frontend

- Software :React.js
- Alasan :React.js adalah library JavaScript populer yang memungkinkan pembuatan UI dinamis dan interaktif dengan komponen-komponen yang mudah dikelola. Berikut adalah alat untuk mempercepat dan mempermudah proses pengembangan React,
 - Tailwind CSS (framework)
 - Vite (build tool modern)

- Library:

Dependencies (Untuk Production)

- **@fullcalendar** → Menampilkan jadwal dengan tampilan grid harian, mingguan, atau bulanan.
- **cors** → Mengaktifkan Cross-Origin Resource Sharing untuk memungkinkan komunikasi antara frontend dan backend di domain berbeda.
- **qrcode** → Digunakan untuk membuat QR Code dari data tertentu, misalnya untuk fitur tambahan berbasis kode QR.
- **react-big-calendar** → Library kalender lain yang mendukung tampilan jadwal dalam berbagai format (mingguan, harian, bulanan) untuk project.
- **react-icons** → Ikon untuk digunakan di komponen UI Anda.
- **react-router-dom** → Mengatur navigasi antar halaman.

DevDependencies (Untuk Development)

- **@vitejs/plugin-react**
- **autoprefixer**
- **tailwindcss**
- **vite**

2. Backend

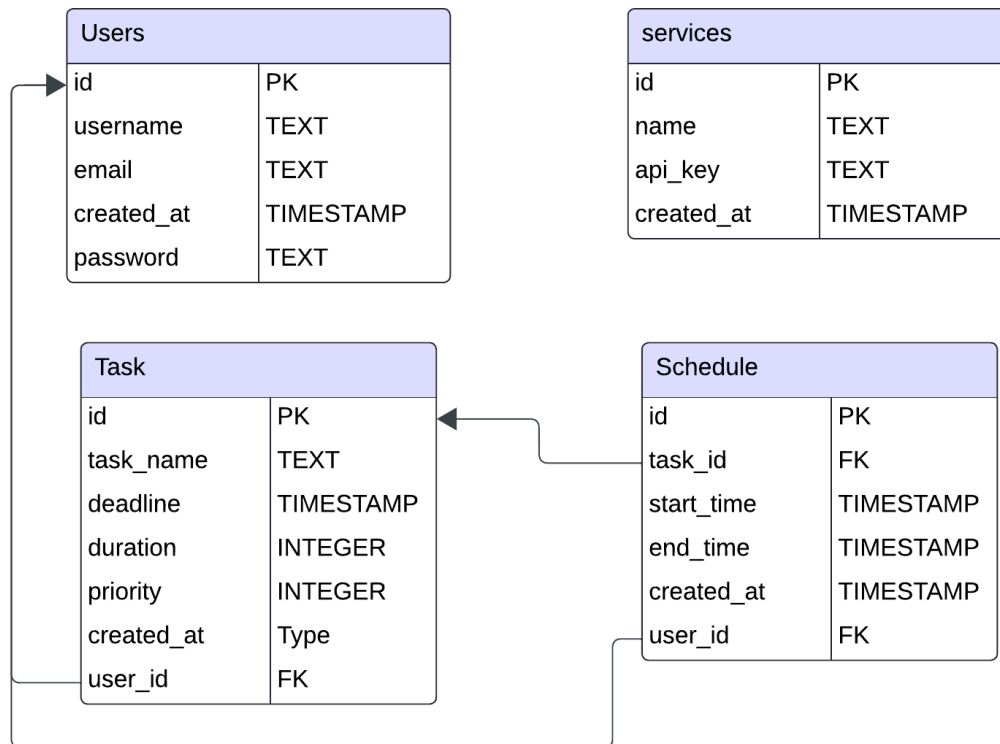
- **Software** : Node.js dengan Express.js
- **Alasan** : Node.js sangat cocok untuk aplikasi yang membutuhkan respons cepat dan dapat menangani banyak permintaan secara bersamaan.
- **Library** :

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

```
"dependencies": {
  "@supabase/supabase-js": "^2.47.10",
  "axios": "^1.7.9",
  "bcrypt": "^5.1.1",
  "cors": "^2.8.5",
  "crypto": "^1.0.1",
  "dotenv": "^16.4.7",
  "express": "^4.21.2",
  "jsonwebtoken": "^9.0.2",
  "passport": "^0.7.0",
  "passport-google-oauth20": "^2.0.0",
  "uuid": "^11.0.4"
},
"devDependencies": {
  "@ljharb/tsconfig": "^0.2.3",
  "typescript": "^5.7.2"
}
}
```

3. Penyimpanan Data

- Software : Supabase (PostgreSQL)
- Alasan :
 - LocalStorage memungkinkan pengguna menyimpan informasi lokal dan mengaksesnya bahkan saat sedang offline untuk tahap development.
 - PostgreSQL adalah sistem database relasional open-source dengan dukungan untuk JSON, indexing, dan transaksi.
- Skema Diagram:



4. Algoritma Penjadwalan

- **Software** : *Greedy Algorithm*

Greedy algorithm adalah algoritma **khusus** yang dikustomisasi oleh Taskly untuk mengatur penjadwalan sesuai dengan kebutuhan bisnis yang ada.

- **Alasan** :

Kriteria	Greedy Algorithm
Definisi	Memecahkan masalah langkah demi langkah dengan memilih solusi terbaik pada saat itu (lokal optimum).
Fokus	Menghasilkan solusi cepat dengan hasil mendekati optimal.
Metode	Digunakan untuk masalah sederhana atau yang dapat diselesaikan secara iteratif.
Waktu eksekusi	O(n) : Hanya satu iterasi melalui variabel (relatif cepat).

5. Visualisasi

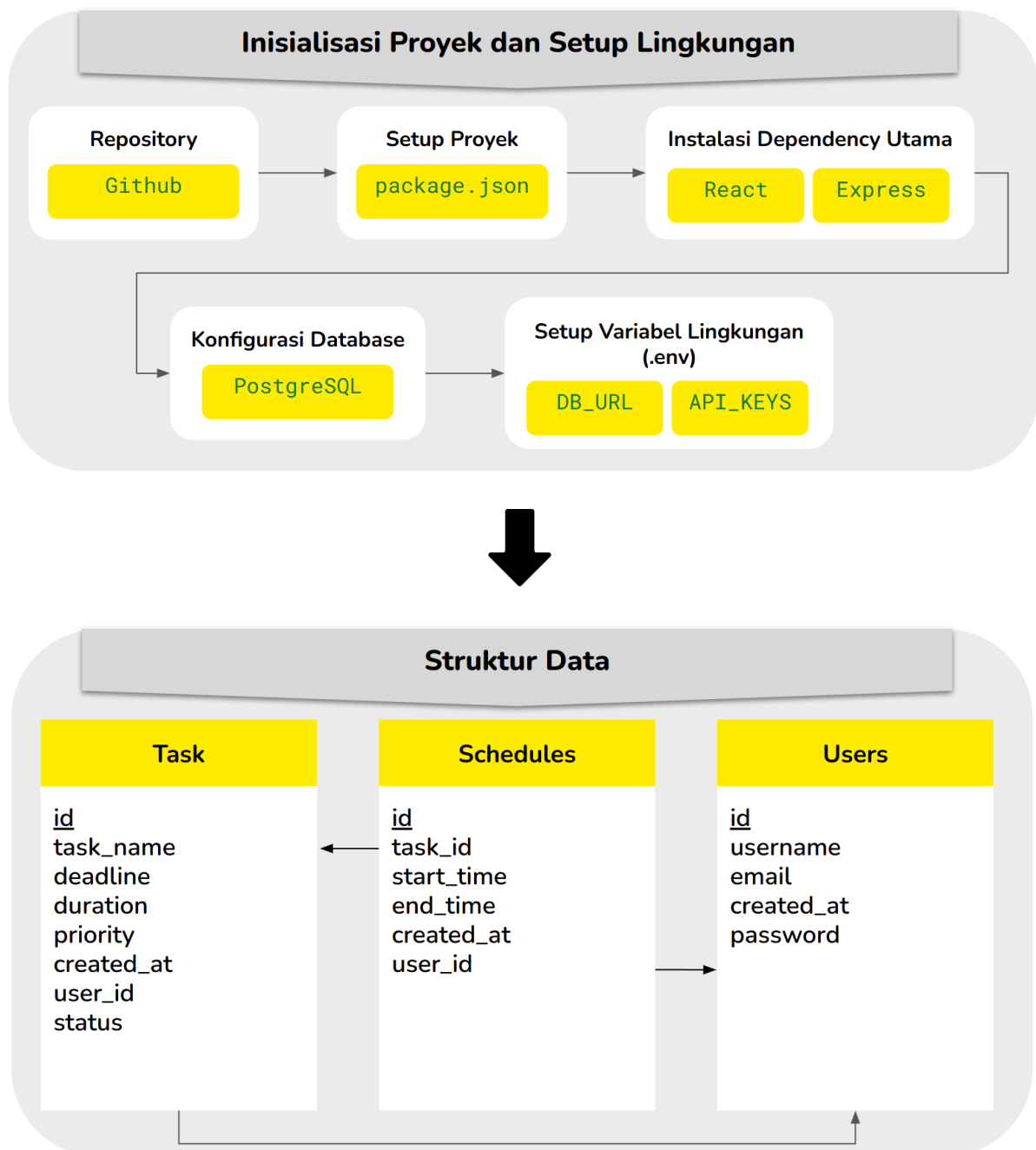
- Kalender
 - Software : React
 - Alasan : tampilan yang lebih familiar dan mudah untuk diintegrasikan. API ini bisa memberikan akses langsung ke tampilan kalender Google yang sudah dikenal luas.
- Grafik Gantt
 - Modul : React Big Calendar
 - Alasan : D3.js adalah library JavaScript yang sangat fleksibel untuk membuat grafik berbasis data. Cocok untuk membuat grafik Gantt yang dapat disesuaikan dengan kebutuhan aplikasi.

6. Hosting dan Deployment

- Software : Biznet VPS Gio CCloud
- Alasan : Performa tinggi, keamanan, dan skalabilitas dengan latensi rendah, cocok untuk aplikasi yang membutuhkan stabilitas dan efisiensi.
- Requirement:
 - VPS Configuration:
 - CPU: 1 vCPU
 - RAM: 1 GB
 - Storage: 60 GB Disk
 - Domain Name untuk menghubungkan aplikasi Anda dengan URL yang mudah diakses.
 - Public IP Address untuk mengakses server secara publik.
 - SSL Certificate untuk mengamankan koneksi melalui HTTPS.
 - Software Requirements:
 - Node.js untuk menjalankan backend.
 - Nginx atau Apache sebagai web server.

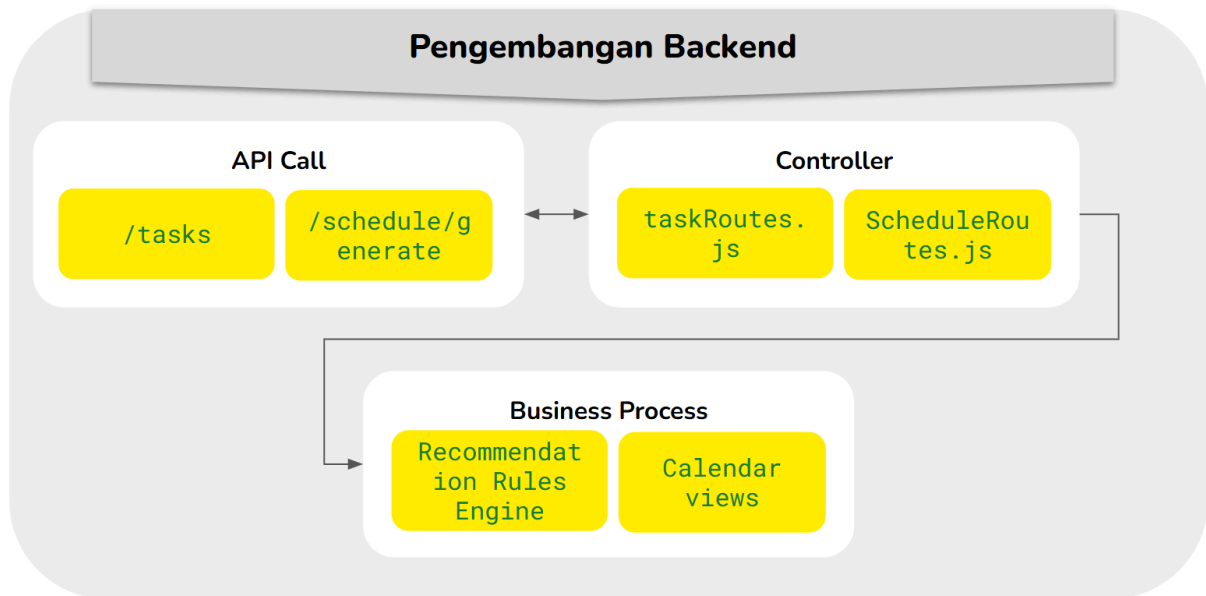
- Supabase sebagai database.
- Access:
 - SSH key untuk akses aman ke server.
 - Tools seperti scp atau rsync untuk transfer file.

D. Langkah Implementasi

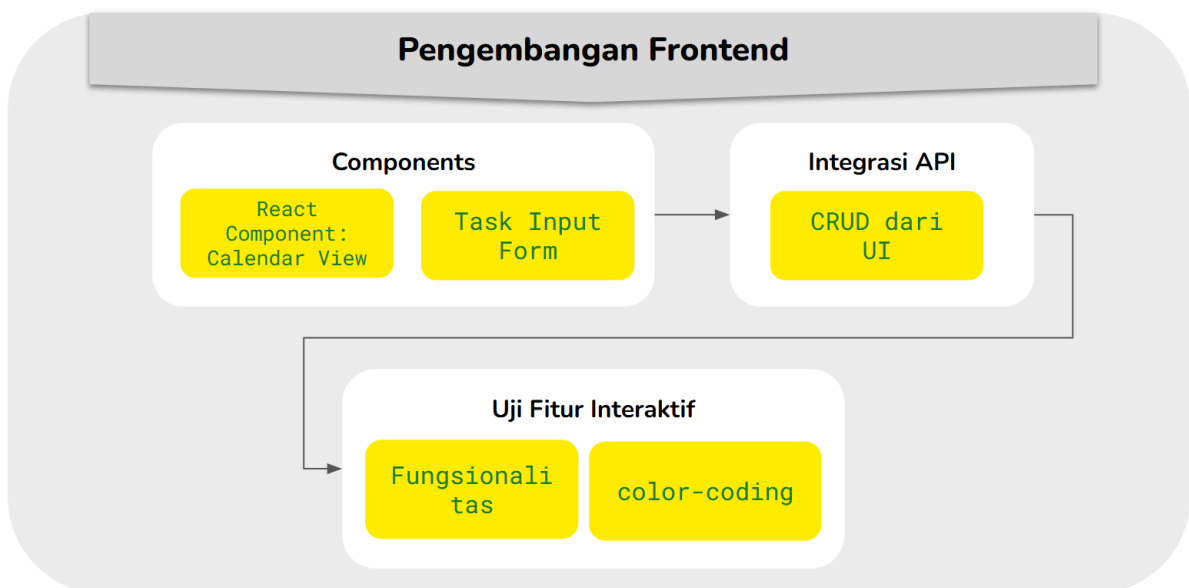


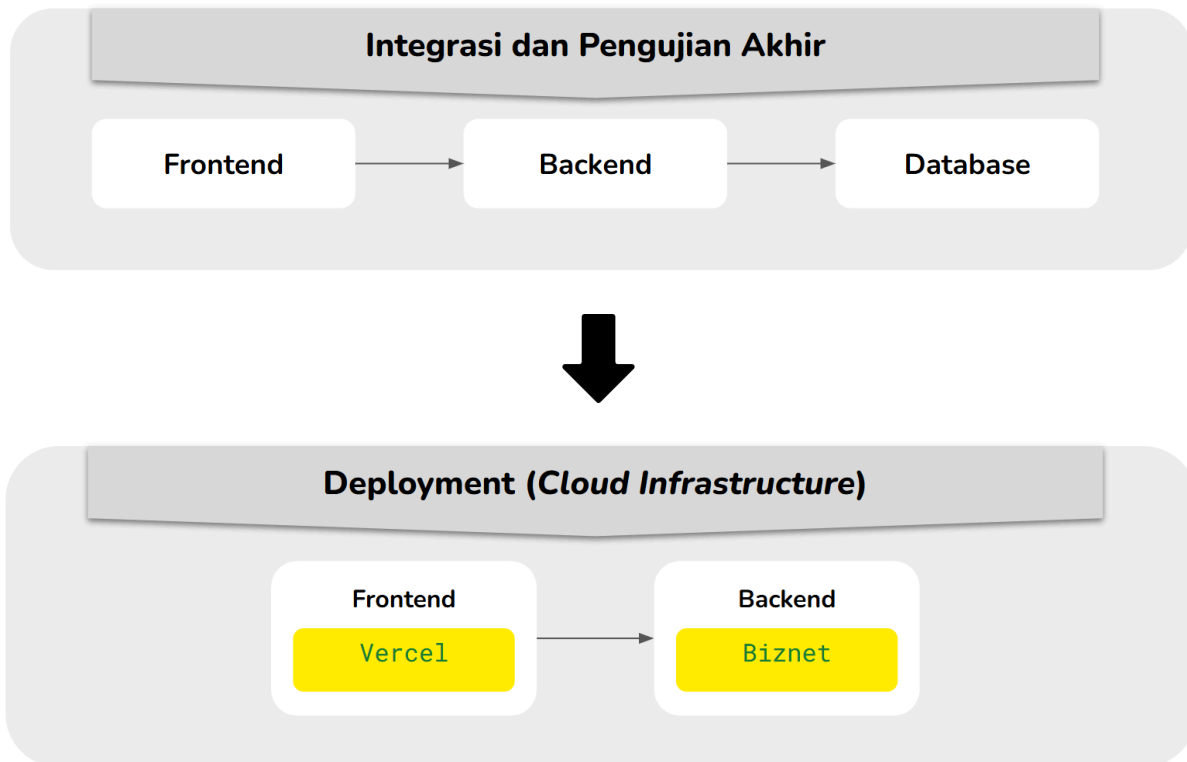


Pengembangan Backend



Pengembangan Frontend





E. Fitur

Repository:

Dalam Taskly, fitur authentication yang diimplementasikan mencakup:

- **Sign Up**
Pengguna dapat membuat akun baru dengan memasukkan data email (unik), username, dan password (minimal 8 karakter).
- **Login**
Pengguna dapat masuk ke dalam sistem menggunakan email dan password yang telah didaftarkan.
- **Logout**
Pengguna dapat keluar dari sistem untuk menjaga keamanan akun mereka.
- **Task**
Pengguna bisa menambahkan (satu maupun sekaligus banyak), menghapus, ataupun mengedit task yang sudah ditambahkan.
- **Payment**
Merupakan integrasi dengan Service Payment sehingga pada masing-masing task pengguna bisa membeli rekan yang bisa mengerjakan tugas tersebut.
- **Schedule**
Pengguna bisa membuat schedule secara otomatis dari hasil input

yang sudah ditambahkan dan melihat hasilnya dalam bentuk calendar.

F. Certificate HTTPS

Frontend https://www.taskly.web.id	Backend https://api.taskly.web.id
<div> <div>Certificate Viewer: *.taskly.web.id</div> <div> <div>General</div> <div>Details</div> </div> <div> <p>Issued To</p> <p>Common Name (CN) *.taskly.web.id</p> <p>Organization (O) <Not Part Of Certificate></p> <p>Organizational Unit (OU) <Not Part Of Certificate></p> <p>Issued By</p> <p>Common Name (CN) R10</p> <p>Organization (O) Let's Encrypt</p> <p>Organizational Unit (OU) <Not Part Of Certificate></p> <p>Validity Period</p> <p>Issued On Thursday, January 9, 2025 at 3:02:03 PM</p> <p>Expires On Wednesday, April 9, 2025 at 3:02:02 PM</p> <p>SHA-256 Fingerprints</p> <p>Certificate 4ddc39ac9117bb382e4da0aaf7468b614a29af66e0d22963d36207296e87fd98</p> <p>Public Key 5f1e6b1bcea0a7d3e37235c28b9b67c778366acecbf953f9b2fb751b834ddbc9</p> </div> </div>	<div> <div>Certificate Viewer: api.taskly.web.id</div> <div> <div>General</div> <div>Details</div> </div> <div> <p>Issued To</p> <p>Common Name (CN) api.taskly.web.id</p> <p>Organization (O) <Not Part Of Certificate></p> <p>Organizational Unit (OU) <Not Part Of Certificate></p> <p>Issued By</p> <p>Common Name (CN) R11</p> <p>Organization (O) Let's Encrypt</p> <p>Organizational Unit (OU) <Not Part Of Certificate></p> <p>Validity Period</p> <p>Issued On Thursday, January 9, 2025 at 7:50:24 PM</p> <p>Expires On Wednesday, April 9, 2025 at 7:50:23 PM</p> <p>SHA-256 Fingerprints</p> <p>Certificate 949cd3cc5c55d8a0594a1f1cdd164e96d78cc2390afef0d90679c780771a8a9f</p> <p>Public Key 3f5512b9d2e038c864ccc43c3b2c9efa03d3f67bbc187260aade32007d5d3362</p> </div> </div>

G. Integrasi Service

1. Menggunakan Service

- Service
 - Nama service : Solstra Payments
 - Developer : **Muhammad Faiz Atharrahman (18222063)**
 - Alasan integrasi : Dibutuhkan tutor untuk membantu selama mengerjakan tugas sehingga API ini cukup dibutuhkan
- Endpoint Solana

```
API Key:
https://api-staging.solstra.fi/user/auth/google

# Create Payment
Request:
POST https://api-staging.solstra.fi/service/pay/create
{
```

```

    "currency": "SOL" || "USDT" || "USDC",
    "amount": 0.00001
  }
}
Response:
{
  status: "success",
  message: "New Payment Order Created",
  data: {
    id: paymentID,
    currency: "SOL" || "USDT" || "USDC",
    amount: "0.0001",
    walletAddress:
"7Msyw4gn6Gp4AASeg3vVzaLMfgE524rQwxYBGwmTETMT",
    checkPaid:
"https://api-staging.solstra.fi/service/pay/${paymentID}/check"
  }
}

# Check Payment
Request:
POST https://api-staging.solstra.fi/service/pay/${paymentID}/check
Response:
{
  "status": "success",
  "message": "Payment Status: Unpaid",
  "data": {
    "id": "677d3262bcf6b1016a804602",
    "currency": "SOL",
    "amount": 0.00001,
    "walletAddress":
"7Msyw4gn6Gp4AASeg3vVzaLMfgE524rQwxYBGwmTETMT",
    "checkPaid":
"https://api-staging.solstra.fi/service/pay/${paymentID}/check",
    "isPaid": false,
    "isExpired": false
  }
}

```

• Hasil Implementasi Taskly

```

// Create Payment
https://api.taskly.web.id/tasks/payment

```

```
// Check Payment
https://api.taskly.web.id/tasks/status

Body:
{
  "paymentID": <ID-Payment>
}
```

2. Menyediakan Service

- Service yang menggunakan Taskly
 - a. Nama service : Piano Progression
 - b. Developer : *Nasywaa Anggun Athiefah (18222021)*
 - c. Alasan integrasi : Dibutuhkan variasi dari algoritma penjadwalan, contohnya *greede algorithm* yang dimiliki oleh Taskly

- Endpoint Taskly

Otentikasi
<p>URL: /protected/signup</p> <p>Method: POST</p> <p>Deskripsi: Mendaftarkan layanan baru dan mendapatkan API Key untuk otentikasi.</p> <p>Headers:</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "name": "<Nama-Service>" }</pre>
Endpoint ini akan mengembalikan api key untuk username yang dimasukkan
Service
<p>URL: /schedule/external</p> <p>Method: POST</p> <p>Deskripsi: Membuat jadwal berdasarkan tugas yang dikirimkan oleh layanan eksternal.</p> <p>Headers:</p> <p>x-api-key: <></p> <p>Content-Type: application/json</p>

Body:

```
{
  "tasks": [
    {
      "task_name": "Task A",
      "deadline": "2025-01-05T10:00:00Z"
    }
  ]
}
```

Endpoint ini akan mengembalikan schedule yang sudah terjadwalkan.

H. Kesimpulan

Taskly adalah platform otomatisasi penjadwalan yang dirancang untuk membantu pengguna mengelola tugas dengan lebih efisien. Menggabungkan teknologi seperti React.js, Node.js, dan Supabase, Taskly menghadirkan fitur-fitur utama seperti jadwal otomatis, manajemen tugas, dan dilengkapi juga dengan integrasi pembayaran.

Dengan algoritma Greedy yang cepat dan mendekati optimal, ditambah visualisasi yang *user-friendly*, Taskly memberikan pengalaman yang ringkas dan *powerful* untuk mengatur waktu dengan lebih baik. Platform ini punya potensi besar untuk mendukung produktivitas penggunanya.

Dalam tahap implementasi beberapa saya mendapatkan banyak pelajaran. Mulai dari menyusun struktur file backend dan frontend, hingga melakukan deployment. Hal paling *challenging* yang saya rasakan adalah melakukan deployment backend menggunakan Bisnet. Dari pengalaman ini saya mempelajari banyak hal, di antaranya,

- Mengatur routing
- Menggunakan file docker untuk deployment
- DNS server
- Penggunaan postman untuk testing, dan masih banyak lagi.