

HR Service API Documentation - Job Description & Case Study

Job Description - Product Engineer (Backend) 2025

Perusahaan: Rakamin is hiring a Product Engineer (Backend) to work on Rakamin. We're looking for dedicated engineers who write code they're proud of and who are eager to keep scaling and improving complex systems, including those powered by AI.

About the Job:

You'll be building new product features alongside a frontend engineer and product manager using our Agile methodology, as well as addressing issues to ensure our apps are robust and our codebase is clean. This role touches on building AI-powered systems, designing and orchestrating how large language models (LLMs) integrate into Rakamin's product ecosystem.

Contoh Pekerjaan:

- Collaborating with frontend engineers and 3rd parties to build robust backend solutions.
- Developing and maintaining server-side logic for central database, ensuring high performance.
- Designing and fine-tuning AI prompts.
- Building **LLM chaining flows**.
- Implementing **Retrieval-Augmented Generation (RAG)** by embedding and retrieving context from vector databases.
- Handling **long-running AI processes gracefully** including job orchestration, async background workers, and retry mechanisms .
- Designing safeguards for uncontrolled scenarios: managing failure cases and mitigating the **randomness/nondeterminism of LLM outputs**.
- Writing reusable, testable, and efficient code.
- Strengthening our test coverage with RSpec.

Requirements / About You:

- Strong track record on **backend technologies** of web apps, ideally with exposure to **AI/LLM development** or a strong desire to learn.
- Experience with backend languages and frameworks (**Node.js, Django, Rails**).
- Modern backend tooling and technologies:
 - Database management (MySQL, PostgreSQL, MongoDB).
 - RESTful APIs.
 - Cloud technologies (AWS, Google Cloud, Azure).
 - Server-side languages (Java, Python, Ruby, or JavaScript).
 - Scalable application design principles.
 - Familiarity with **LLM APIs, embeddings, vector databases** and **prompt design best practices**.

Lokasi:

Remote, only looking for people based in **Indonesia** .

Case Study Brief - Inti Logika & Deliverables

Objective:

Build a backend service that automates the initial screening of a job application.

Core Logic & Data Flow - Inputs to be Evaluated:

1. Candidate CV (PDF).
2. Project Report (PDF).

Deliverables - Backend Service (API Endpoints):

- **POST /upload**: Accepts Candidate CV & Project Report (multipart/form-data), stores files, returns document IDs .
- **POST /evaluate**: Triggers **asynchronous** AI evaluation pipeline, receives job title and document IDs, immediately returns a job ID .
- **GET /result/{id}**: Retrieves status ("queued", "processing", "completed") and final result .

Deliverables - Evaluation Pipeline Components:

1. **RAG (Context Retrieval)**: Ingest all System-Internal Documents into a vector database, retrieve relevant sections, and inject into prompts .
2. **Prompt Design & LLM Chaining**: Pipeline consists of CV Evaluation, Project Report Evaluation, and Final Analysis synthesis .
3. **Long-Running Process Handling**: POST /evaluate must not block. Use job ID to track status .
4. **Error Handling & Randomness Control**: Simulate/handle LLM API failures (timeouts, rate limits), implement **retries/back-off**, control LLM temperature .

Project Report Deliverable Parameters (fokus Project):

- Correctness (meets requirements: prompt design, chaining, **RAG**, handling errors).
- Code Quality (clean, modular, testable).
- Resilience (handles failures, retries).
- Documentation (clear README, explanation of trade-offs).
- Creativity / Bonus (optional improvements like authentication, deployment).

Scoring Rubric for Case Study Evaluation

CV Match Evaluation (1-5 scale per parameter)

Parameter	Description	Scoring Guide	Weight
Technical Skills Match	Alignment with job requirements (backend, databases, APIs, cloud, AI/LLM).	1=Irrelevant skills, 4=Strong match, 5=Excellent match + AI/LLM exposure.	40%
Experience Level	Years of experience and project complexity.	1=<1 yr, 3=2-3 yrs with mid-scale projects, 5=5+ yrs / high-impact projects.	25%
Relevant Achievements	Impact of past work (scaling, performance, adoption).	1=No clear achievements, 3=Some measurable outcomes, 5=Major measurable impact.	20%
Cultural/Collaboration Fit	Communication, learning mindset, teamwork/leadership.	1=Not demonstrated, 3=Average, 5=Excellent and well-demonstrated.	15%

Project Deliverable Evaluation (1-5 scale per parameter)

Parameter	Description	Scoring Guide	Weight
Correctness (Prompt & Chaining)	Implements prompt design, LLM chaining, RAG context injection.	1=Not implemented, 4=Works correctly, 5=Fully correct + thoughtful.	30%
Code Quality & Structure	Clean, modular, reusable, tested.	1=Poor, 4=Good structure + some tests, 5=Excellent quality + strong tests.	25%
Resilience & Error Handling	Handles long jobs, retries, randomness, API failures.	1=Missing, 4=Solid handling, 5=Robust, production-ready.	20%
Documentation & Explanation	README clarity, setup instructions, trade-off explanations.	1=Missing, 4=Clear, 5=Excellent + insightful.	15%
Creativity / Bonus	Extra features beyond requirements.	1=None, 4=Strong enhancements, 5=Outstanding creativity.	10%

Overall Evaluation Output:

- **CV Match Rate:** Weighted Average (1-5) → Convert to 0-1 decimal $(\times 0.2)$.
- **Project Score:** Weighted Average (1-5).
- **Overall Summary:** 3-5 sentences (strengths, gaps, recommendations).