

BCM - Baka!! Chuunibyou Mode



Don't misunderstand! It's not like I'm modelling business capability for you or anything...

“Business Capability Modelling”

Ditulis Oleh

M. Rayhan Farrukh

13523035

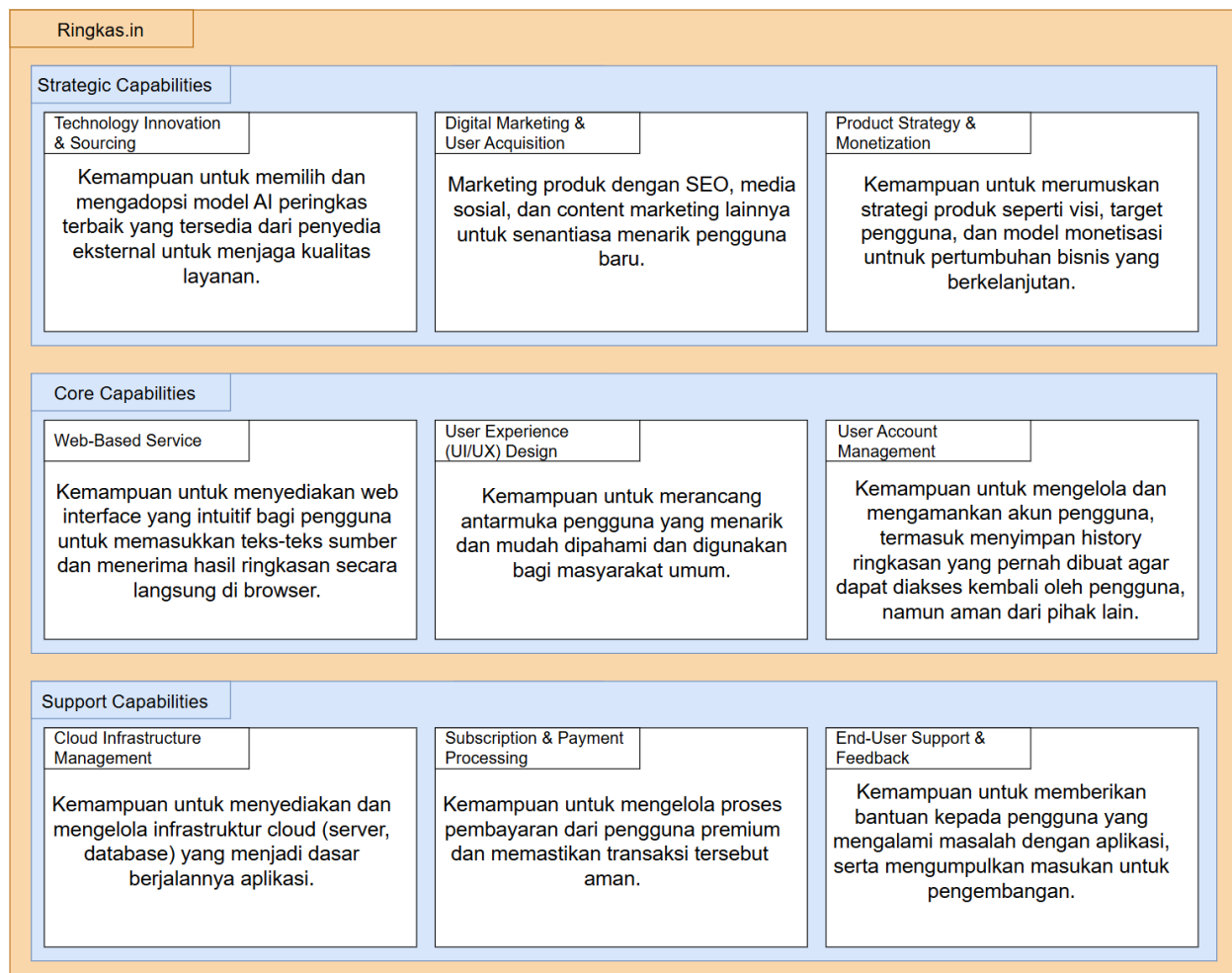
A. Business Capability Model

Nama Bisnis: Ringkas.in

Deskripsi: Memberikan layanan pembuat ringkasan otomatis, yang dapat menerima banyak teks sumber dari pengguna dan membuat ringkasan dari teks-teks tersebut.

Model Bisnis: Business to Client (B2C)

Business Capaility Model



Penjelasan Diagram

1. Strategic Capabilities

Kemampuan strategis yang dirancang untuk memenangkan persaingan pasar, serta mempertahankan pertumbuhan jangka panjang.

- *Technology Innovation and Sourcing*, menjaga kualitas layanan agar tetap unggul sehingga mempertahankan retensi pengguna

- *Digital Marketing and User Acquisition*, senantiasa mendapatkan pengguna baru.
- *Product Strategy and Monetization*, menentukan arah pengembangan dan cara menghasilkan pendapatan.

II. *Core Capabilities*

Kemampuan utama yang memberikan *value* secara langsung ke pengguna.

- *Web-based Service*, menyediakan fungsi utama layanan peringkasan otomatis yang mudah diakses oleh pengguna.
- *User Experience Design*, membuat antarmuka layanan utama mudah dan enak untuk digunakan oleh semua pengguna.
- *User Account Management*, mengelola identitas dan data pengguna secara aman, agar integritas data pengguna aman.

III. *Support Capabilities*

Kemampuan internal perusahaan untuk membantu memastikan kelancaran operasional layanan pada aplikasi.

- *Cloud Infrastructure Management*, memastikan infrastruktur teknis berjalan dengan andal
- *Subscription and Payment Processing*, mengelola proses transaksi secara aman, efisien, dan simpel bagi pengguna.
- *End-User Support and Feedback*, memberikan bantuan kepada pengguna serta mengumpulkan *feedback* untuk pengembangan produk sesuai yang diinginkan pengguna.

B. Analisis Kebutuhan Teknologi

Untuk memenuhi kebutuhan bisnis, berikut teknologi-teknologi yang dibutuhkan Ringkas.in.

- *Framework* aplikasi - Next.js (*front-end*) & NestJS (*back-end*)

Next.js dipilih untuk *front-end* karena sudah menjadi standar untuk aplikasi web modern. Fitur-fitur yang tersedia pada Next.js akan membuat aplikasi Ringkas.in menjadi lebih berkualitas.

NestJS dipilih sebagai *back-end* karena sifatnya yang sangat modular akan cocok dengan aplikasi Ringkas.in yang akan menggabungkan beberapa layanan dengan logika bisnis berbeda.

- Infrastruktur Data - Amazon RDS & Redis

Amazon RDS for PostgreSQL digunakan sebagai basis data, dengan fitur otomasi untuk tugas-tugas seperti *backup* dan *recovery*, sehingga menjaga integritas data.

Lalu untuk meningkatkan performa ringkasan, akan digunakan sistem *caching* dengan Redis. Redis dipilih karena lebih *feature rich* dibandingkan layanan *caching* lain.

- *Authorization & Authentication* - OAuth 2.0 & OpenID Connect

OAuth 2.0 digunakan sebagai protokol otorisasi untuk mengamankan pengguna dan aplikasi, dan OpenID Connect sebagai lapisan autentikasi yang berjalan di atas OAuth 2.0.

- Arsitektur - *Microservices*

Aplikasi akan berbentuk *microservices* untuk memisahkan fungsi bisnis, sehingga setiap layanan dapat dikelola sesuai kebutuhannya.

- *API Gateway* - Amazon API Gateway

Amazon *API Gateway* dipilih sebagai pengatur *traffic* antar *microservices* pada aplikasi, yang menerapkan otorisasi, *rate limiting*, dan *logging* API.

- *Deployment & DevOps* - AWS, Docker & Amazon EKS

AWS digunakan sebagai *cloud platform* utama untuk memenuhi kebutuhan infrastruktur yang *scalable* dan *high-availability*.

Docker untuk *containerization* agar aplikasi dapat berjalan konsisten dalam berbagai lingkungan.

Amazon EKS akan bertindak sebagai manajer otomatis *container* aplikasi dengan fitur seperti *auto-scaling* dan *load balancing* agar memudahkan skalabilitas aplikasi.

- **Model AI - OpenAI, Anthropic, MistralAI**
Menggunakan kombinasi beberapa API LLM yang dipilih sesuai *membership tier* dan preferensi pengguna, untuk menyeimbangkan kualitas, kecepatan pemrosesan, dan biaya operasional.
- ***Payment processor* - Stripe**
Stripe dipilih sebagai *payment processor* untuk memenuhi kebutuhan *user* lokal maupun internasional.
- ***Monitoring dan Logging* - Datadog**
Menggunakan Datadog untuk memantau performa sistem, mengumpulkan *log* aplikasi dan melakukan analisis ketika terjadi sebuah insiden, agar dapat ditangani dengan cepat.