**SOFTWARE REQUIREMENT SPECIFICATION**

**[Website Jakarta.go,id]**

# DAFTAR ISI

[**DAFTAR ISI**](#_wl0fo79wm8du) **1**

[**RIWAYAT DOKUMEN**](#_xbl48irlnhmv) **3**

[**PENDAHULUAN**](#_gscs2av0mnww) **4**

[Objektif Dokumen](#_tmccrrnbz5s9) 4

[Target Audiens](#_qnr3vduiuomx) 4

[Referensi](#_6l8ysbriqfw1) 4

[**ARSITEKTUR PROYEK/PRODUK/SISTEM/APLIKASI/FITUR**](#_7o8170ekd50m) **5**

[Objektif](#_z8y2qi9m4122) 5

[Kerangka Layanan](#_jhf8xqyyg1ra) 5

[Technology Stack](#_97zatusfddl) 5

[Application Environment](#_ahebthy6so3b) 6

[Database Environment](#_fdtitt3siust) 7

[Engineer/Developer](#_425ladocbaqj) 8

[Lainnya](#_aaxkbnt46qkv) 9

[Batasan Pengembangan dan Implementasi](#_blwrq5a3jyct) 10

[**RINCIAN TEKNIS**](#_wni5hrczy4d9) **11**

[Objektif](#_s6gw2kxcmpth) 11

[Akses Repository](#_q951718k5612) 11

[Struktur Database](#_1sc8y5vt3u5k) 11

[Dokumentasi Sistem](#_e9yvjos3031) 11

[Modul/Fitur #1](#_iuouxhwiltzy) 11

[Modul/Fitur #2 dan seterusnya](#_vhmq0ubdhlz) 12

[Dokumentasi API](#_98oz1v7ymlkj) 12

[Istilah dan Pengertian](#_eg0sq6bqk7m5) 12

[Bagian-Bagian Dokumentasi API](#_6w6c9rlfcljk) 12

[Request](#_t5m7awsrq7qo) 12

[Response](#_5nuelmihquwd) 12

[Jenis-jenis Tipe Data yang Digunakan](#_5uqyq9efbtiu) 12

[Kode Status HTTP](#_b4iemnmqnrf2) 13

[Kode Otorisasi](#_lulchn2cs6k) 13

[Spesifikasi API Internal](#_ujfnfx246m3d) 13

[Spesifikasi API Eksternal](#_pj5w0eds8u1) 13

[Pseudocode Struktur Request dan Response](#_dxcxder2hb30) 13

[Dokumentasi API Spesifikasi API Internal](#_5oaf3z8ayfgu) 13

[API #1](#_s0byx8w6bzkc) 13

[API #2 dan seterusnya](#_tvxbbay3fs2a) 14

[Dokumentasi API Spesifikasi API Eksternal](#_65r52brq7ldm) 14

[API #1](#_5c9791rmfmtj) 14

[API #2 dan seterusnya](#_2ymx8s2e3el9) 14

# RIWAYAT DOKUMEN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Tanggal** | **Versi** | **Keterangan** | **Nama** |
|  |  |  |  |  |
|  |  |  |  |  |

# PENDAHULUAN

## Objektif Dokumen

*Software Requirement Specification* (SRS) atau dapat diartikan Spesifikasi Kebutuhan Perangkat Lunak (SKPL), adalah suatu dokumen yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu *software*.

Pada dasarnya SRS adalah suatu dokumen yang menyatakan kebutuhan perangkat lunak sebagai hasil dari proses analisis yang dilakukan dalam konteks pengembangan perangkat lunak. Komponen-komponen dari proyek/produk/sistem/aplikasi/fitur [nama] yang akan dijelaskan dalam dokumen ini mencakup penjabaran *technology stack* yang digunakan, lokasi *repository* tempat *source code* berada, struktur database, penjabaran dari fungsi-fungsi beserta relasinya yang ada di dalam suatu fitur/modul/menu, dan dokumentasi API.

Dokumen ini dimaksudkan agar dapat menjadi fleksibel akan perubahan sepanjang pengembangan proyek/produk/sistem/aplikasi/fitur, karena dalam tahap pengerjaannya akan diselesaikan dalam berbagai *sprint* dan rilis. Bukan tidak mungkin akan banyak sekali perubahan fungsi dan komponen utama di dalamnya. Untuk saat ini, tujuan utama pembuatan dokumen ini adalah untuk mengidentifikasi persyaratan teknis/non-fungsional pada tingkat *low-level*.

## Target Audiens

Dokumen ini ditujukan kepada *Software Architect*, dan *Software Engineer/Developer* agar dapat menyamakan persepsi dan menyetujui persyaratan untuk pengembangan dan penerapan proyek/produk/sistem/aplikasi/fitur [nama].

## Referensi

Cantumkan dokumen atau link lain yang dirujuk untuk dapat melengkapi informasi mengenai proyek/produk/sistem/aplikasi/fitur [nama] ini. Termasuk ke dalamnya dokumen paparan, BRD, FSD, *test case*, *manual book*, link Design (Figma), link Wiki, atau dokumen eksternal lainnya yang terkait.

Berikan informasi yang cukup sehingga pembaca dapat mengakses salinan setiap referensi, termasuk judul, penulis, nomor versi, tanggal, sumber, dan lokasi.

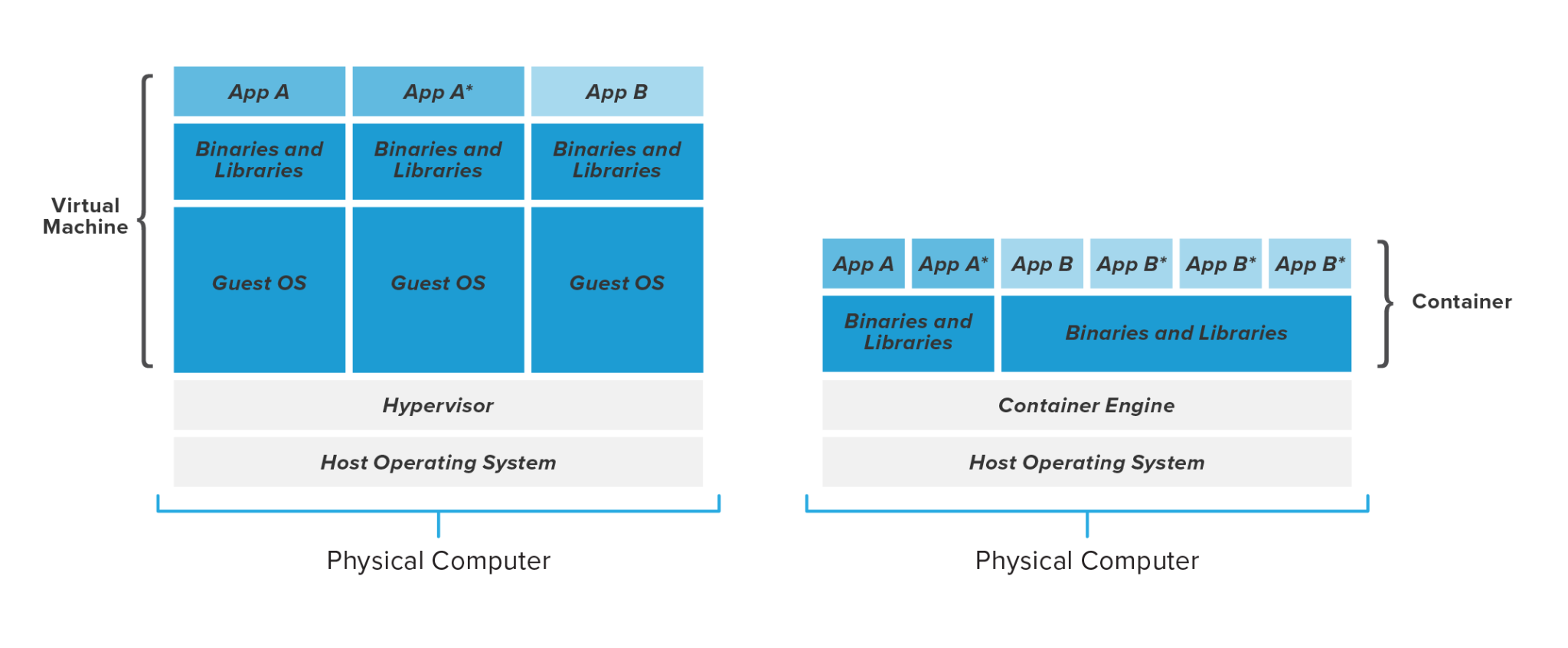
# ARSITEKTUR PROYEK/PRODUK/SISTEM/APLIKASI/FITUR

## Objektif

Pada bagian ini, akan dijelaskan gambaran umum mengenai arsitektur dari proyek/produk/sistem/aplikasi/fitur [nama] dalam bentuk diagram sederhana yang menggambarkan lapisan layanan beserta *technology stack* yang digunakan.

## Kerangka Layanan

Gambarkan lapisan layanan yang akan diterapkan pada pengembangan proyek/produk/sistem/aplikasi/fitur [nama] dalam bentuk diagram sederhana. Contoh:



## Technology Stack

Sebutkan technology stack yang digunakan untuk pengembangan proyek/produk/sistem/aplikasi/fitur [nama]. Direkomendasikan berbentuk tabel seperti contoh di bawah ini:

### *Application Environment*

|  |  |  |
| --- | --- | --- |
| **Development** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP Local |  |  |
| IP Public |  |  |
| Domain/Subdomain |  |  |
| **Staging** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP Local |  |  |
| IP Public |  |  |
| Domain/Subdomain |  |  |
| **Production** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP Local |  |  |
| IP Public |  |  |
| Domain/Subdomain |  |  |

### *Database Environment*

|  |  |  |
| --- | --- | --- |
| **Development** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP |  |  |
| Port |  |  |
| RDMS |  |  |
| **Staging** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP |  |  |
| Port |  |  |
| RDMS |  |  |
| **Production** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Type |  |  |
| OS |  |  |
| CPU |  |  |
| RAM |  |  |
| Storage |  |  |
| IP |  |  |
| Port |  |  |
| RDMS |  |  |

### *Engineer/Developer*

|  |  |  |
| --- | --- | --- |
| **DevOps Engineer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Packaging |  |  |
| Orchestrator |  |  |
| GUI |  |  |
| Configurator |  |  |
| Language |  |  |
| Web Service |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |
| **Fullstack Developer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Framework |  |  |
| Language |  |  |
| Map |  |  |
| Map Library |  |  |
| Map Language |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |
| **Backend Developer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Framework |  |  |
| Language |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |
| **Frontend Developer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Framework |  |  |
| Language |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |
| **Android Developer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Framework |  |  |
| Language |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |
| **iOS Developer** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Framework |  |  |
| Language |  |  |
| Rekomendasi *Tools* (*Hardware* dan *Software*) Pengembangan |  | |

### Lainnya

|  |  |  |
| --- | --- | --- |
| ***Tools* (Hardware/Software)** | | **Keterangan (Versi) & Konfigurasi Lainnya** |
| Email SMTP |  |  |
| SMS Gateway |  |  |
| ... |  |  |

### 

## Batasan Pengembangan dan Implementasi

Jelaskan masalah apa pun yang akan membatasi pengembangan proyek/produk/sistem/aplikasi/fitur [nama] ataupun pada implementasinya. Hal Ini mungkin termasuk: kebijakan perusahaan atau peraturan; keterbatasan perangkat keras; antarmuka ke aplikasi lain; teknologi tertentu, *tools*, atau database yang akan digunakan untuk misalkan; operasi paralel; protokol komunikasi; keamanan; konvensi desain atau standar pemrograman.

# RINCIAN TEKNIS

## Objektif

Pada bagian ini, akan dijelaskan alur akses ke dalam *repository* proyek/produk/sistem/aplikasi/fitur [nama], struktur database, struktur modul/fitur beserta fungsi-fungsi yang ada di dalamnya, dan juga dokumentasi/kontrak APInya.

## Akses *Repository*

Jabarkan alur lokasi repository dari proyek/produk/sistem/aplikasi/fitur [nama] secara runut.

## Struktur Database

Gambaran jelas mengenai struktur database dari proyek/produk/sistem/aplikasi/fitur [nama].

## Dokumentasi Sistem

Penjelasan singkat mengenai proyek/produk/sistem/aplikasi/fitur [nama] dan sebutkan modul/fitur utama apa saja yang ada di dalamnya.

### Modul/Fitur #1

Penjelasan singkat mengenai modul/fitur #1. Sebutkan Classes, Objects, Methods, atau Functions Utama yang ada di dalamnya berikut dengan penjelasan singkatnya.

|  |  |
| --- | --- |
| **Class/Object/ Method/Function** | **Deskripsi** |
| Nama #1 |  |
| Nama #2 |  |
| Nama #3 |  |
| ... |  |

### Modul/Fitur #2 dan seterusnya

## Dokumentasi API

Application Programming Interface (API) memungkinkan developer aplikasi untuk mengintegrasikan/menggunakan data/informasi atau bertukar data/informasi dari aplikasi lain untuk digabungkan dengan aplikasi yang akan dikembangkan.

Penjelasan tambahan yang lebih spesifik terkait API proyek/produk/sistem/aplikasi/fitur [nama].

### Istilah dan Pengertian

Dibuatkan dalam bentuk tabel saja biar rapi.

### Bagian-Bagian Dokumentasi API

REST API (*Representational State Transfer Application Programming Interface*) yang digunakan pada dokumen ini menggunakan bagian-bagian berikut untuk mendeskripsikan parameter dan/atau apapun yang dibutuhkan untuk melakukan *request* serta mendapatkan *response* dari proses *request* sebelumnya.

Bagian-bagian itu terdiri dari:

#### Request

Penjabaran penjelasan di sini, jika ada daftar/list yang ingin disebutkan, baiknya dalam bentuk tabel agar terlihat lebih ringkas.

#### Response

Penjabaran penjelasan di sini, jika ada daftar/list yang ingin disebutkan, baiknya dalam bentuk tabel agar terlihat lebih ringkas.

### Jenis-jenis Tipe Data yang Digunakan

Dibuatkan dalam bentuk tabel saja biar rapi.

|  |  |
| --- | --- |
| **Tipe Data** | **Deskripsi** |
| #1 |  |
| #2 |  |
| #3 |  |
| ... |  |

### Kode Status HTTP

Dibuatkan dalam bentuk tabel saja biar rapi.

|  |  |
| --- | --- |
| **Kode Status HTTP** | **Deskripsi** |
| #1 |  |
| #2 |  |
| #3 |  |
| ... |  |

### Kode Otorisasi

Dibuatkan dalam bentuk tabel saja biar rapi.

|  |  |
| --- | --- |
| **Kode Otorisasi** | **Deskripsi** |
| #1 |  |
| #2 |  |
| #3 |  |
| ... |  |

### Spesifikasi API Internal

Dibuatkan dalam bentuk tabel saja biar rapi.

### Spesifikasi API Eksternal

Dibuatkan dalam bentuk tabel saja biar rapi.

### *Pseudocode* Struktur *Request* dan *Response*

Untuk menyederhanakan penjelasan dari data yang akan dikirimkan ke server atau dikembalikan dari server maka kita menggunakan *pseudocode* dari bentuk format YAML (Yet Another Markup Language) dibandingkan bentuk tabular agar lebih sesuai dengan struktur yang akan dikomunikasikan. Untuk lebih jelas apa itu YAML, dapat dilihat pada URL <https://en.wikipedia.org/wiki/YAML>.

### Dokumentasi API Spesifikasi API Internal

API ini digunakan untuk keperluan manajemen data serta informasi dari dan ke dalam proyek/produk/sistem/aplikasi/fitur [nama].

#### API #1

#### API #2 dan seterusnya

### Dokumentasi API Spesifikasi API Eksternal

#### API #1

#### API #2 dan seterusnya

### 