

## Perancangan Aplikasi Manajemen Pengelolaan Data Jemaat Berbasis *Microservices Website*

Michael Septiadi <sup>1\*</sup>, Intan Purnamasari <sup>2</sup>, Carudin <sup>3</sup>

<sup>1,2,3</sup> Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, Kabupaten Karawang, Provinsi Jawa Barat, Indonesia

### article info

#### Article history:

Received 17 May 2023

Received in revised form

9 August 2023

Accepted 12 September 2023

Available online October 2023

#### DOI:

<https://doi.org/10.35870/jti.k.v7i4.1195>

#### Keywords:

Microservices Website; SDLC  
Waterfall; Whitebox Testing.

#### Kata Kunci:

Microservices Website; SDLC  
Waterfall; Whitebox Testing.

### abstract

Data management that is still done manually at God's DNA Cut Mutiah Church makes the data team have to do data collection manually as well. Hence, this makes the data team have to work extra by collecting new data needed directly. Errors such as inputting data incorrectly and the like may occur. This study aims to solve these problems by designing microservices-based data management application websites. Using the SDLC waterfall methodology can help facilitate the design of this application. The SDLC waterfall methodology has five stages: analysis, design, implementation, testing, and maintenance. In its implementation, to build microservices-based applications, this website is designed using MongoDB for databases, ReactJS at the front, and FastAPI Python at the back. This application has been tested using the white-box testing method, and the results of the test run according to the design and needs determined.

### abstrak

Pengelolaan data yang masih dilakukan secara manual pada Gereja God's DNA Cut Mutiah membuat tim data harus melakukan pendataan secara manual pula, sehingga hal tersebut membuat tim data harus bekerja secara ekstra dengan mengumpulkan data-data baru yang dibutuhkan secara langsung. Kesalahan-kesalahan seperti salah menginputkan data dan semacamnya mungkin saja dapat terjadi. Penelitian ini bertujuan untuk memberikan solusi dari permasalahan tersebut dengan melakukan perancangan aplikasi manajemen pengelolaan data berbasis *microservices* website. Dengan menggunakan metodologi SDLC waterfall dapat membantu memudahkan dalam perancangan aplikasi ini. Dalam prakteknya, metodologi SDLC waterfall ini memiliki 5 tahapan yaitu analysis, design, implementation, testing, dan maintenance. Pada implementasinya, untuk membangun aplikasi berbasis *microservices* website ini dirancang dengan menggunakan MongoDB untuk database, ReactJS sebagai frontend, dan FastAPI Python sebagai backend. Aplikasi ini telah diuji dengan menggunakan metode whitebox testing dan hasil dari pengujian tersebut berjalan sesuai dengan perancangan dan kebutuhan yang ditentukan.

\* Author. Email: [septiadimichael@gmail.com](mailto:septiadimichael@gmail.com) <sup>1\*</sup>, [intan.purnamasari@staff.unsika.ac.id](mailto:intan.purnamasari@staff.unsika.ac.id) <sup>2</sup>, [carudin@staff.unsika.ac.id](mailto:carudin@staff.unsika.ac.id) <sup>3</sup>.

© E-ISSN: 2580-1643.

Copyright @ 2023. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET)  
(<http://creativecommons.org/licenses/by/4.0/>).

## 1. Latar Belakang

Dalam era digital yang terus berkembang, penggunaan teknologi informasi telah menjadi pondasi utama untuk menjalankan berbagai kegiatan sehari-hari. Sistem-sistem digital memungkinkan pelaksanaan tugas dan proses secara lebih efisien, efektif, cepat, dan akurat. Pemanfaatan teknologi informasi tidak hanya membentuk cara kita bekerja, berkomunikasi, dan berinteraksi, tetapi juga memperkenalkan kemungkinan-kemungkinan baru dalam pengelolaan data dan pemrosesan informasi. Penerapan teknologi informasi menawarkan sejumlah keuntungan signifikan, terutama dalam konteks pengelolaan data. Proses pengumpulan, penyimpanan, dan pemrosesan data dapat dilakukan secara lebih terstruktur dan otomatis, mengurangi risiko kesalahan manusia dan meningkatkan akurasi informasi yang dihasilkan. Keberadaan teknologi informasi bukan hanya mempermudah pelaksanaan tugas, tetapi juga memberikan solusi untuk meningkatkan produktivitas dan kualitas pekerjaan.

Rumah ibadah, sebagai bagian integral dari masyarakat, tidak terkecuali dalam memanfaatkan teknologi informasi. Salah satu tantangan yang kerap dihadapi adalah pendataan yang masih dilakukan secara manual oleh tim data. Proses ini rentan terhadap potensi kesalahan dalam pencatatan data, yang dapat mempengaruhi integritas dan keakuratan informasi di rumah ibadah tersebut. Oleh karena itu, perlu adanya solusi yang mengintegrasikan teknologi informasi untuk meningkatkan efisiensi pengelolaan data di lingkungan rumah ibadah. Penelitian ini bertujuan memberikan solusi terhadap permasalahan tersebut melalui analisis kebutuhan dan pengamatan mendalam terkait pengelolaan data di rumah ibadah. Data yang diperoleh dari analisis ini menjadi dasar pengambilan keputusan dalam merancang solusi yang efektif. Tujuan utama penelitian ini mencakup penyediaan aplikasi atau sistem yang terintegrasi dengan database, guna memfasilitasi tim pendataan dalam pengelolaan data rumah ibadah. Selain itu, penelitian ini juga bertujuan untuk mempermudah tugas tim pendataan dalam proses pengumpulan data. Solusi yang diimplementasikan dirancang untuk meminimalkan potensi kesalahan manusia dan meningkatkan efisiensi dalam pengumpulan data. Diharapkan, hasil penelitian ini dapat memberikan

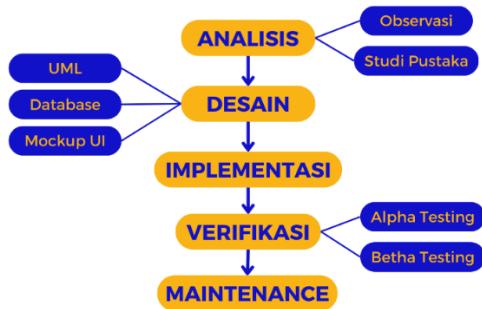
kontribusi positif terhadap efektivitas pengelolaan data di rumah ibadah, mengurangi risiko kesalahan, dan meningkatkan kualitas informasi yang dihasilkan.

Dalam penelitian ini, konsep perancangan aplikasi manajemen pengelolaan data berbasis *microservices* website memperoleh landasan dari berbagai penelitian terkait yang telah dilakukan sebelumnya. Dalam penerapan arsitektur *microservices*, Putra *et al.* (2021) merancang sistem pengantaran (delivery) untuk PT. Jejaring Hijau Indonesia, menunjukkan bahwa pendekatan *microservices* dapat memberikan solusi yang efektif dalam pengelolaan sistem perusahaan [1]. Seiring dengan itu, Adhi (2021) melaporkan perancangan dan pembangunan website konfesi online dengan menerapkan konsep *microservices*, menekankan manfaat arsitektur ini dalam pengembangan aplikasi web [2]. Selain itu, Lombogia *et al.* (2022) menyajikan penelitian tentang perancangan arsitektur perangkat lunak *microservices* pada aplikasi open library Universitas Telkom dengan menggunakan gRPC, menunjukkan fleksibilitas dan ketangguhan arsitektur *microservices* [3]. Begitu juga, Ridwan *et al.* (2021) merancang marketplace berbasis website menggunakan metodologi Systems Development Life Cycle (SDLC) dengan model waterfall, menggambarkan penerapan metode pengembangan yang terstruktur dalam konteks *microservices* [4]. Pentingnya perancangan arsitektur *microservices* dalam berbagai domain juga ditekankan oleh Elsen (2021) dalam rancang bangun portal berita daring, menyoroti keunggulan dan adaptabilitas arsitektur ini dalam konteks portal berita online [5]. Dengan mempertimbangkan penelitian-penelitian terkait tersebut, penelitian ini memiliki tujuan mendalam untuk menghadirkan solusi yang terkini dan relevan dalam perancangan aplikasi manajemen pengelolaan data di rumah ibadah. Diharapkan, hasil penelitian ini dapat memperkaya pemahaman kita terhadap potensi penggunaan teknologi informasi, khususnya konsep *microservices*, dalam mengatasi tantangan pengelolaan data di berbagai konteks organisasi.

## 2. Metode Penelitian

Penelitian ini dilakukan dengan menggunakan metodologi *System Development Life Cycle* yang

tahapannya dijelaskan sebagai berikut dan pada Gambar 1 merupakan gambaran alur dari setiap tahapannya.



Gambar 1. Alur Penelitian

- 1) Analisis, pada tahapan ini dilakukan observasi permasalahan yang terjadi pada rumah ibadah tersebut dan melakukan analisis kebutuhan pengguna dan sistem.
- 2) Desain, tahapan ini merupakan tahapan dilakukannya perancangan desain dari hasil analisis yang telah didapatkan dari tahapan seluruhnya. Pada tahapan desain ini terbagi ke dalam 3 tahap desain, yaitu desain sistem, desain website, dan desain *database*.
- 3) Implementasi, dilakukan dengan menggunakan *framework* ReactJS dan FastAPI Python untuk pengimplementasianya.
- 4) Pengujian atau *Testing*, tahapan ini terbagi menjadi 2 kali tahapan pengujian setelah aplikasi dibangun dengan *Alpha Testing* menggunakan *Whitebox Testing* dan *Betha Testing*.
- 5) Perawatan atau *Maintenance*, tahap ini dilakukan dengan tujuan mengembangkan sistem atau aplikasi yang sudah ada agar fungsi pada sistem yang telah dibuat dapat berjalan dengan optimal.

Metode penelitian dilakukan dengan melakukan observasi ke objek penelitian untuk menentukan permasalahan yang terjadi, perancangan sistem yang akan dibuat. Pada tahap percancangan desain menggunakan *Unified Modeling Language* (UML) untuk desain sistem, desain *database*, dan desain aplikasi dengan bantuan *tools* Figma. Sedangkan, pada tahap perancangan sistem pada penelitian ini akan menggunakan arsitektur *microservices website*, dimana pada implementasinya arsitektur *microservices* ini terbagi menjadi 2 bagian, yaitu *frontend* dan *backend*.

Selanjutnya, dilakukan pengujian untuk melakukan pengecekan fungsi-fungsi pada aplikasi. Pada tahap perawatan, aplikasi sudah digunakan dan pada tahap ini dapat dilakukan pengembangan sistem apabila terdapat fungsi-fungsi yang perlu diperbaharui.

### 3. Hasil dan Pembahasan

#### *Analisis*

Pada penelitian ini dilakukan analisis untuk mengidentifikasi masalah-masalah yang terjadi pada objek yang diteliti. Pengidentifikasi masalah dilakukan dengan observasi untuk melihat secara langsung permasalahan yang terjadi. Dari hasil observasi yang dilakukan ada beberapa masalah yang melatarbelakangi penelitian ini sebagai berikut.

- 1) Pendataan yang dilakukan secara manual apabila terdapat jemaat baru yang belum terdata oleh tim data. Hal tersebut membuat tim data perlu menanyakan secara langsung biodata dari jemaat baru dan menyimpan data ke dalam google sheet.
- 2) Permintaan dokumen atau permohonan baptis yang dibutuhkan oleh jemaat dilakukan dengan meminta secara langsung atau face-to-face kepada tim data untuk dibuatkan permintaan dokumen tersebut, dengan begitu pendataan yang dilakukan menjadi tidak efektif.
- 3) Perlu adanya aplikasi yang dapat memudahkan pekerjaan dan meminimalisir kesalahan seperti input data secara manual yang dilakukan oleh tim data. Sehingga pendataan jemaat baru dan permintaan dokumen dapat dilakukan oleh jemaat tersebut dengan memasukkan biodata pada aplikasi yang dibuat melalui smartphone ataupun sejenisnya.

Dari 3 permasalahan diatas, dapat disimpulkan bahwa yang menjadi sebuah permasalahan utama penelitian ini, yaitu diperlukannya sebuah sistem atau aplikasi yang dapat menunjang kebutuhan pendataan. Sehingga pendataan sudah tidak lagi dilakukan secara manual, tetapi data tersimpan langsung ke *database* yang sudah terhubung dengan sistem. Dari hasil analisis yang telah dilakukan, dapat ditentukan kebutuhan-kebutuhan dari permasalahan tersebut. Di bawah ini adalah tabel kebutuhan pengguna dengan masing-masing fungsinya.

Tabel 1. Kebutuhan Pengguna

No	Peran	Fungsi
1	Administrator	Mengelola data jemaat dan <i>volunteers</i> God's DNA Cut Mutiah Bekasi Mengelola data <i>Request Document</i> dari jemaat Melakukan pencarian data pada <i>dashboard</i> Menyetujui data <i>New Comers</i> Melihat jadwal ibadah pada halaman website
2	User Pengunjung Website	Mengisi form <i>New Comers</i> untuk jemaat baru Mengisi form <i>Request Document</i> bagi yang membutuhkan

Selain adanya kebutuhan pengguna, ada pula kebutuhan fungsional yang akan dirancang pada penelitian ini yang akan diterapkan ke dalam sistem sebagai berikut.

- 1) Sistem dapat menampilkan jadwal ibadah dan informasi tentang organisasi.
- 2) Sistem dapat melakukan login ke halaman dashboard untuk administrator.
- 3) Sistem dapat menerima input data New Comers.
- 4) Sistem dapat menerima input data Request Document.
- 5) Sistem dapat mengelola data Jemaat dan Request Document.
- 6) Sistem dapat melakukan pencarian data Jemaat dan Request Document.
- 7) Sistem dapat menampilkan grafik data dari database.

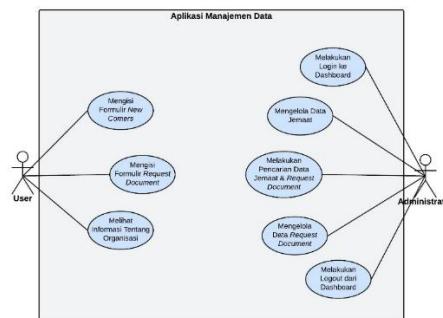
### Desain

Setelah tahapan analisis selesai, selanjutnya masuk ke dalam tahap desain untuk melakukan perancangan dari hasil analisis yang telah didapatkan dari tahap sebelumnya. Pada tahap desain terbagi menjadi 3 tahapan sebagai Berikut.

#### 1) Desain Sistem

Pada tahap pendesainan sistem, perancangan ini dibuat dengan menggunakan *use case diagram*

dari *Unified Modeling Language* atau UML. Penggunaan *use case diagram* ini dapat memudahkan pendefinisian keperluan sistem setelah melakukan analisis. Maka dari itu pada penelitian ini *use case diagram* dibuat untuk mendeskripsikan setiap fungsi yang dibutuhkan. Di bawah ini terdapat gambar *use case diagram* yang dirancang pada penelitian ini.



Gambar 2. Use Case Diagram

#### 2) Desain Database

Pada tahapan selanjutnya adalah melakukan perancangan desain *database* untuk diterapkan pada sistem. *Database* yang digunakan pada penelitian ini menggunakan MongoDB, yang mana *database* ini tidak berbentuk tabel melainkan berbentuk *JavaScript Object Notation* (JSON). Berikut struktur *database* pada perancangan sistem penelitian ini.

```
{
  "id": "String",
  "name": "String",
  "birth_date": "Date",
  "phone_number": "String",
  "status": "String",
  "type": "String",
  "location": "String",
  "created_at": "Date"
}
```

Gambar 3. Struktur JSON Data Jemaat

```
{
  "id": "String",
  "name": "String",
  "category": "String",
  "phone_number": "String",
  "desc": "String",
  "status": "String",
  "created_at": "Date"
}
```

Gambar 4. Struktur JSON Data Request Document

```
{
  "id": "String",
  "username": "String",
  "password": "String"
}
```

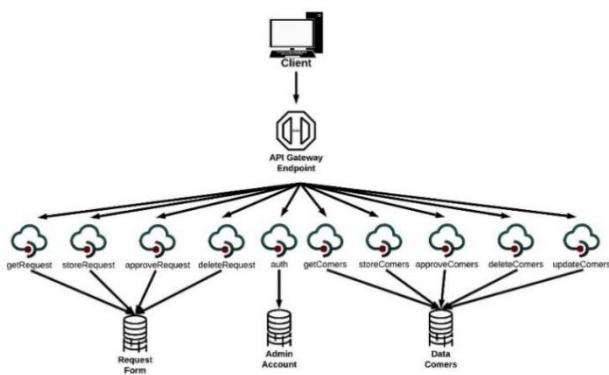
Gambar 5. Struktur JSON Data Akun Admin

### 3) Desain Website

Pada tahapan desain website ini bertujuan untuk memudahkan pada tahap implementasi atau pengkodean pada *frontend*. Berikut ini adalah hasil desain tampilan antarmuka pengguna website yang akan dirancang.

#### *Implementasi*

Setelah tahapan desain selesai, langkah selanjutnya yaitu tahap implementasi atau pengkodean. Pengimplementasian desain ke dalam kode program menggunakan *framework* ReactJS untuk *frontend* dan FastAPI Python untuk *backend* dengan menggunakan bantuan *tools* Visual Studio Code serta menggunakan DBMS MongoDB. Pada tahap ini dihasilkan website yang telah diimplementasikan dari desain. Website yang telah dibuat ini memiliki tampilan *Landing Page*, *Dashboard*, dan API dari fitur-fitur yang dibutuhkan pada website ini. Dengan menerapkan arsitektur *microservices* pada aplikasi ini, maka arsitektur dirancang dengan memisahkan setiap *services* atau layanan yang satu dengan lainnya. Pada Gambar dan Tabel di bawah ini terdapat bentuk arsitektur yang digunakan pada aplikasi ini.

Gambar 6. Arsitektur *Microservices*Tabel 2. Layanan *Microservices*

Kebutuhan <i>Microservices</i>	Method	URL Endpoint
Melakukan <i>login</i> ke <i>dashboard</i> oleh admin	<i>Post</i>	/auth
Menampilkan data <i>request</i>	<i>Get</i>	/getRequest
Menambah		
data <i>request</i>	<i>Post</i>	/storeRequest
Menyetujui		
data <i>request</i>	<i>Put</i>	/approveRequest/{id}
Menghapus		
data <i>request</i>	<i>Delete</i>	/deleteRequest/{id}
Menampilkan data <i>comers</i>	<i>Get</i>	/getComers
Menambah	<i>Post</i>	/storeComers
Menghapus	<i>Delete</i>	/deleteComers/{id}
Mengubah	<i>Put</i>	/updateComers/{id}
Menyetujui	<i>Put</i>	/approveComers/{id}
data <i>comers</i>		

#### *Pengujian/Testing*

Pada tahap pengujian ini bertujuan untuk melakukan uji coba terhadap masing-masing fungsi yang terdapat pada aplikasi yang telah dirancang dan juga untuk mengetahui apakah terdapat kesalahan pada aplikasi. Tahap pengujian ini terdapat 2 pengujian, yaitu *alpha testing* dan *beta testing* dengan hasil pengujian sebagai berikut.

##### 1) *Alpha Testing*

Di tahap ini, pengujian dilakukan dengan menggunakan teknik *blackbox* untuk pengujinya. Pengujian dilakukan untuk memastikan setiap fungsi yang ada pada aplikasi sudah berjalan sesuai dengan perancangan. Di bawah ini terdapat hasil pengujian *alpha testing* dengan menggunakan teknik *blackbox* pada penelitian ini.

##### 2) *Beta Testing*

Tahap *beta testing* dilakukan dengan cara pengguna

menggunakan sistem secara langsung untuk di uji coba apakah sistem yang dibuat mudah digunakan atau tidak. Pada tahap ini terdapat 3 pengujii, yaitu 2 admin pengelola data dan 1 *user* pengguna yang bukan admin. Pengujian dilakukan dengan memberikan instruksi atau perintah untuk diselesaikan oleh pengujii. Dari hasil pengujian tersebut, ketiga pengujii dapat menyelesaikan instruksi dengan baik sehingga dapat disimpulkan bahwa aplikasi yang dirancang mudah digunakan. Selain itu, para pengujii juga dimintai pendapatnya mengenai aplikasi yang telah dibuat setelah melakukan pengujian sistem. Di bawah ini terdapat tabel pendapat dari pengujii.

#### *Perawatan/Maintenance*

Tahapan yang terakhir setelah selesai dilakukan pengujian sistem yaitu tahap perawatan atau *maintenance*. Tahap ini dilakukan dengan tujuan mengembangkan sistem atau aplikasi yang sudah ada agar setiap fungsi pada sistem yang telah dibuat dapat berjalan dengan optimal. Berikut adalah tahap perawatan yang dilakukan.

- 1) Melakukan pengecekan sistem secara rutin selama satu minggu sekali agar sistem dapat berfungsi dengan baik sesuai dengan kriteria sistem yang sudah dirancang.
- 2) Perbaikan sistem jika dalam program ditemukan kesalahan fungsi atau *bug* yang tidak terdeteksi pada tahap pengujian sistem.
- 3) Pengingkatan sistem dengan melakukan modifikasi terhadap sistem ketika terdapat suatu potensi yang dapat dikembangkan di dalam sistem setelah berjalannya sistem beberapa waktu.

## 4. Kesimpulan dan Saran

Pengembangan aplikasi berbasis *website* dengan model arsitektur *microservices* adalah suatu bentuk aplikasi untuk membantu sebuah rumah ibadah dalam mengelola data-data rumah ibadah tersebut. Perancangan dibangun dengan menggunakan metode SDLC dengan model *waterfall* dengan melewati tahapan analisis kebutuhan sistem, desain sistem, implementasi atau pengkodean, uji coba sistem atau *testing*, dan perawatan sistem atau

*maintenance*. Dengan menggunakan arsitektur *microservices*, membuat website ini menjadi lebih ringan karena setiap fitur yang ada dapat berjalan sendiri tanpa harus bergantung dengan fitur lainnya. Aplikasi ini dapat dikembangkan lagi sesuai dengan kebutuhan sistem dengan menambahkan fitur-fitur baru yang diperlukan. Selain itu, penggunaan API juga dapat dikembangkan ke dalam sistem berbasis *mobile* dengan mengintergrasikan API yang sama pada sistem berbasis website.

## 5. Daftar Pustaka

- [1] Putra, D.S.D., Margita, S. and Subhanna, M., 2021. Perancangan Sistem Delivery Untuk PT. Jejaring Hijau Indonesia Dengan Arsitektur Microservice. *ALGOR*, 2(2), pp.11-22. DOI: <https://doi.org/10.31253/algov2i2.548>.
- [2] Adhi, G.N., 2021. *Perancangan dan Pembangunan Website Konfesi Online Dengan Menerapkan Microservices* (Thesis, Universitas Atma Jaya Yogyakarta).
- [3] Lombogia, J.F., Syahrina, A. and Musnansyah, A., 2022. PERANCANGAN ARSITEKTUR PERANGKAT LUNAK MICROSERVICES PADA APLIKASI OPEN LIBRARY UNIVERSITAS TELKOM MENGGUNAKAN gRPC. *Telkatika: Jurnal Telekomunikasi Elektro Komputasi & Informatika*, 1(2).
- [4] Jurnal, J.T.I.K., 2021. Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall. *Jurnal JTIIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5, p.2.
- [5] Elsen, R., 2021. Perancangan Arsitektur Microservice untuk Portal Berita Daring. *Jurnal Algoritma*, 18(2), pp.352-357. DOI: <https://doi.org/10.33364/algoritma/v.18-2.875>.