

PENGEMBANGAN WEB SERVICE PADA SISTEM INFORMASI JAHE MENGGUNAKAN METODE REST API

Gilang Pujianto, Freza Riana, Safaruddin Hidayat A

Teknik Informatika, Universitas Ibn Khaldun Bogor
Jalan Sholeh Iskandar Kedung Badak Tanah Sareal Kota Bogor
gilang302017@gmail.com

ABSTRAK

Seiring berkembangnya teknologi sistem informasi, pertukaran data dilakukan menggunakan *web service* agar bisa diakses melalui *platform* berbeda. Untuk mengembangkan sistem informasi jahe ini diperlukan pertukaran data melalui *web service* sehingga dapat digunakan untuk mengembangkan sistem informasi berbeda *platform* seperti *mobile application*, maka dibutuhkan *database server* yang terintegrasi dengan *web service*. Tidak hanya itu dibutuhkan juga keamanan untuk datanya menggunakan *Json Web Token (JWT)*. *Web service* mempunyai konsep yang menguntungkan bagi penggunaannya yaitu untuk dapat lebih mudah mengakses data, misalnya dengan adanya *web service* pada sebuah sistem *website* terdapat perubahan data, maka dengan tersistem data yang ada pada aplikasi android secara otomatis juga berubah. Metode yang digunakan pada sistem ini adalah metode *waterfall* dan menggunakan bahasa pemrograman *javascript* dengan *framework Node JS* serta *Express JS*. Hasil penelitian ini menunjukkan bahwa pengembangan *web service* pada sistem informasi jahe dapat menyediakan pertukaran informasi yang akurat serta *real-time* kepada pengguna yang bisa diakses pada *platform* yang berbeda. Pengujian untuk *web service* menggunakan metode *black box* dan menggunakan *software* postman dinyatakan berhasil.

Kata kunci : Tanaman Jahe, Webservice, Metode Waterfall

1. PENDAHULUAN

Indonesia dikenal sebagai negara yang kaya produksi rempah – rempah. Salah satu kekayaan rempah – rempahnya adalah jahe. Komoditas ini sudah dikenal sejak zaman penjajahan belanda. Rimpangan jahe sangat diminati karena manfaatnya yaitu kesehatan, kesegaran, dan campurannya untuk membuat masakan[1].

Tanaman jahe atau yang biasa disebut dengan nama ilmiah *zingiber officinale* merupakan tanaman herbal tegak, tingginya mencapai 40-100 cm, dan merupakan tanaman yang bisa berumur tahunan. Jahe juga dapat disebut dengan tanaman obat hidup[2].

Sistem informasi tanaman jahe yang memiliki website tersendiri merupakan sistem yang berisi informasi jahe, pembudidayaan jahe dan usaha tani jahe yang masih menggunakan database server untuk pengambilan data tentang informasi jahe. Dengan berkembangnya teknologi sistem informasi, pertukaran data dilakukan menggunakan *web service* agar bisa diakses melalui *platform* berbeda. Untuk mengembangkan sistem informasi jahe ini diperlukan pertukaran data melalui *web service* sehingga dapat digunakan untuk mengembangkan sistem informasi berbeda *platform* seperti *mobile application* maka dibutuhkan database server yang terintegrasi dengan *web service*. Tidak hanya itu dibutuhkan juga keamanan untuk datanya menggunakan *Json Web Token (JWT)*. Menerapkan *web service* dengan teknologi *JSON web token (JWT)* ini akan mengubah proses autentikasi untuk meningkatkan keamanan data sistem informasi. *JWT* adalah token berupa string *JSON* yang sangat padat ukurannya dan menghasilkan

pertukaran informasi dengan melakukan autentikasi[3].

Web service mempunyai konsep yang menguntungkan bagi penggunaannya yaitu untuk dapat lebih mudah mengakses data, misalnya dengan adanya *web service* pada sebuah sistem *website* terdapat perubahan data, maka dengan tersistem data yang ada pada aplikasi android secara otomatis juga berubah. Dengan adanya *API* yang dibuat pengguna lain tidak dapat secara langsung mengakses database. melainkan mereka harus request melalui internet kemudian mengakses *API* dan kemudian menerima respon *API*[4].

Berdasarkan permasalahan diatas, maka penelitian ini akan dijelaskan mengenai bagaimana mengimplementasikan *web service* dan autentikasi keamanan *JWT (Json Web Token)* pada sistem informasi tanaman jahe menggunakan metode *REST API*. Maka penulis akan melakukan penelitian yang berjudul "Pengembangan Web Service Pada Sistem Informasi Jahe Menggunakan Metode REST API".

2. TINJAUAN PUSTAKA

Penelitian sebelumnya terkait pembuatan *JWT* untuk keamanan login pernah dilakukan oleh Ficry dkk dimana pada penelitian tersebut membahas tentang implementasi *JWT* untuk keamanan login. Dan mendapatkan hasil "*signature verified*" dikarenakan token yang dihasilkan selalu berubah dan tidak mengubah isi *header*[3].

Penelitian lainnya pernah dilakukan juga oleh Ramos dkk tentang pembuatan sistem informasi pelatihan berbasis web menggunakan teknologi *web service* untuk pertukaran data. Hasil pengujian yang

didapatkan dinyatakan layak secara fungsional dan dapat menghasilkan *output* yang diharapkan[6].

2.1. Web Service

Web service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi dan interoperabilitas antar sistem dalam jaringan. *Web service* digunakan sebagai dasar untuk memberikan layanan (dalam bentuk informasi atau data) ke sistem lain, sehingga mereka dapat berinteraksi dengan sistem melalui layanan yang disediakan. *Web service* menyimpan data informasi dalam format *JSON* atau *XML* sehingga sistem ini dapat diakses oleh sistem lain, meskipun mereka memiliki platform, sistem operasi, dan bahasa pemrograman yang berbeda [6]

2.2. Application Programming Interface

Application Programming Interface (API) adalah sekumpulan fungsi, sub program, protokol komunikasi, atau alat untuk membuat perangkat lunak. Dengan adanya *API* ini, memudahkan programmer untuk membongkar perangkat lunak dan kemudian mengembangkannya atau mengintegrasikan dengan perangkat lunak lain [7].

2.3. API Gateway

API Gateway adalah titik masuk sistem, merutekan permintaan ke layanan mikro sesuai yang sesuai, memanggil layanan mikro, dan menggabungkan hasilnya. Menyediakan *API* yang disesuaikan untuk setiap klien untuk meneruskan permintaan, mengubah protokol, dan menerapkan logika umum seperti autentikasi dan pembatasan kecepatan. Itu juga dapat bertanggung jawab untuk berbagi tugas seperti otentikasi, pemantauan, dan pemrosesan respons statis. Dalam beberapa kasus *API Gateway* juga bertindak sebagai penyeimbang beban karena mengetahui lokasi dan alamat semua layanan [8].

2.4. Representation State Transfer

Representation State Transfer biasa disingkat dengan (*REST*), *REST* merupakan arsitektur *web service* yang berevolusi dari beberapa gaya arsitektur berbasis jaringan yang biasa diterapkan pada *web services*. Arsitektur *REST* biasanya menjalankan melalui *HTTP (Hypertext Transfer Protocol)*, yang melibatkan proses membaca halaman web tertentu yang berisi *file XML* atau *JSON*. Setiap permintaan adalah independent, *server* tidak menghostingnya, status permintaan apapun. *Application Programming Interface (API)* yang sesuai dengan gaya *REST* disebut *RESTful API*. *RESTful API* menggunakan *Uniform Resource Identifier (URI)* untuk mewakili sumber daya. Setiap sumber data diidentifikasi menggunakan asosiasi *URI*. Metode yang digunakan dalam *REST* meliputi: *GET* untuk mendapatkan sumber daya, *POST* digunakan untuk membuat sumber daya baru, dan metode *PUT* untuk memperbarui sumber daya berdasarkan sumber daya. Sedangkan metode

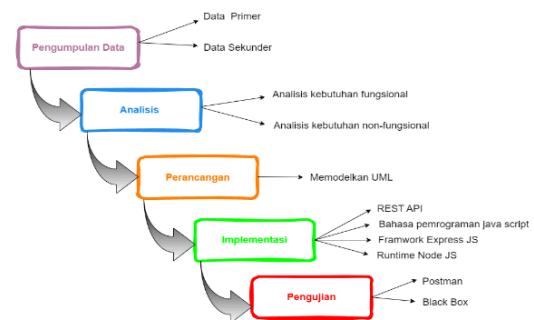
DELETE digunakan untuk menghapus suatu *resource* atau sekelompok *resource* [10].

2.5. Json Web Token

JWT ini adalah *token* berupa string *JSON* sangat padat (ukurannya), informasi mandiri yang digunakan untuk menerapkan sistem autentikasi dan pertukaran informasi, karena ukurannya yang kecil, *token JWT* dapat dikirim melalui *URL*, parameter *HTTP POST* atau di *header HTTP*, dan juga karena ukurannya yang kecil bisa mengirimkan lebih cepat. Ini disebut informasi independen karena konten *token* yang dihasilkan berisi informasi dari permintaan pengguna, jadi tidak perlu melakukan *query* basis data lebih dari satu kali. *Token* bisa diverifikasi dan disetujui karena ditandatangani dengan rahasia (*algoritme HMAC*) atau pasangan kunci publik/pribadi (*algoritma RSA*). Proses login yang dilakukan tidak seperti aplikasi website biasa, tetapi menggunakan *session* untuk mengingat yang sedang melakukan proses *login*. Namun *API* hanya menggunakan konsep *JWT* bisa disebut “*jot*”.[11]

3. METODE PENELITIAN

Metode penelitian yang digunakan untuk penelitian ini adalah metode *Waterfall*, metode ini memiliki 5 tahapan yaitu, pengumpulan data, analisis, perancangan, pengkodean, dan pengujian. Metode penelitian *Waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metode Waterfall

3.1. Pengumpulan Data

Pada tahap pengumpulan data dilakukan untuk memudahkan penelitian dalam mengumpulkan data yang dibutuhkan terkait proses analisis. Adapun data yang dibutuhkan dibagi menjadi dua bagian yaitu:

3.2. Data Primer

Dalam penelitian, peneliti melakukan wawancara langsung dengan pembuat sistem informasi tanaman jahe. Hasil wawancara dapat dilihat pada Tabel 1.

Tabel 1. Data Primer

No	Nama Data	Data
1.	Nama <i>database</i>	Jahe_
2.	<i>Database</i> yang digunakan	<i>Mysql Server</i>
3.	Jumlah tabel	13 tabel

3.3. Data Sekunder

Data sekunder ialah data yang didapat dari sumber seperti jurnal, *e-book*, *website* dan lain – lain. Dengan cara studi literatur terkait dengan, sistem informasi, tanaman jahe, *json web token*, serta pembuatan sistem dengan teknologi *web service*.

3.4. Analisis

Pada tahap ini dilakukan proses analisis dengan tujuan dilakukan untuk mendefinisikan kebutuhan sistem dan menganalisis pengelolaan data yang diterima. Meliputi kebutuhan fungsional sistem, kebutuhan non-fungsional, arsitektur sistem, sistem yang sedang berjalan, dan sistem yang dibangun.

3.5. Perancangan

Pada tahap ini dilakukan perancangan sistem yang di modelkan dalam UML, rancangan *database* yang diperoleh pada tahap analisis, dan rancangan instalasi *web service* menggunakan metode REST API. Perancangan utama dilakukan adalah perancangan *database*, perancangan *use case diagram*, perancangan *class diagram*, perancangan *component diagram*, dan perancangan *deployment diagram*.

3.6. Implementasi

Pada tahap ini menggunakan *framework node JS* dan *Express JS* untuk mendorong proses implementasi dengan cara yang dapat dimengerti mesin menggunakan bahasa pemrograman berbasis *javascript*. Setelah proses implementasi selesai, maka dilakukan proses pengujian sistem secara bertahap.

3.7. Pengujian

Pada tahap ini dilakukan proses pengujian sistem dengan tujuan untuk mengidentifikasi kekurangan pada sistem yang dibangun. Pengujian sistem ini menggunakan metode *black box* untuk memverifikasi fungsi dari setiap *web service* yang dibuat. Pengujian dilakukan dengan menggunakan *software postman*.

3.8. Analisis Kebutuhan

Tahap analisis ini meliputi proses analisis kebutuhan sistem antara lain analisis kebutuhan fungsional sistem, kebutuhan nonfungsional, arsitektur sistem, sistem yang berjalan, sistem yang akan dibuat.

3.9. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah analisis kebutuhan pada sistem berupa layanan *web service* yang disediakan untuk pengelolaan sistem manajemen. Berdasarkan kebutuhan pengelolaan sistem manajemen maka modul – modul yang akan di sediakan oleh *web service*. Adapun modul yang diimplementasikan dapat dikelompokkan sebagai berikut :

- a. Modul Registrasi
Modul Registrasi digunakan admin untuk membuat akun, agar bisa dipakai login kedalam sistem.
- b. Modul Login
Modul login digunakan admin untuk mendapatkan token untuk memverifikasi bahwa admin mendapatkan hak akses untuk bisa masuk kedalam sistem dan bisa mengelola seluruh data informasi tanaman jahe.
- c. Modul Budidaya Jahe
Modul budidaya jahe digunakan untuk mengelola data budidaya, ada beberapa keterangan budidaya seperti kesesuaian iklim, penyiapan benih, penyediaan bibit, pola tanam, pemupukan, penyiapan lahan, panen jahe, dan tahap persemaian. Admin dapat menambah data budidaya jahe, melihat, mengubah dan menghapus data budidaya jahe.
- d. Modul Jenis Jahe
Modul jenis jahe digunakan untuk mengelola data jenis jahe, seperti menambah data jenis jahe, melihat, mengubah, menghapus data jenis jahe.
- e. Modul Kendala Jahe
Modul kendala jahe digunakan untuk mengelola data kendala jahe, seperti kendala budidaya dan kendala usaha tani jahe, admin dapat menambah data kendala jahe, melihat, mengubah, menghapus data kendala jahe.
- f. Modul Menu
Modul menu digunakan untuk mengelola data menu, seperti menambah data menu, melihat, mengubah dan menghapus data menu.
- g. Modul Menu Frontend
Modul menu frontend digunakan untuk mengelola data menu frontend, seperti menambah data menu frontend, melihat, mengubah dan menghapus data menu frontend.
- h. Modul Morfologi Jahe
Modul morfologi jahe digunakan untuk mengelola data morfologi jahe, seperti menambah data morfologi jahe, melihat, mengubah dan menghapus data morfologi jahe.
- i. Modul Pengertian Jahe
Modul pengertian jahe merah digunakan untuk mengelola data pengertian jahe, seperti menambah data pengertian jahe, melihat, mengubah dan menghapus data pengertian jahe.
- j. Modul Sub Menu
Modul sub menu digunakan untuk mengelola data sub menu, seperti menambah data sub menu, melihat, mengubah dan menghapus data sub menu.
- k. Modul Taksonomi
Modul Taksonomi digunakan untuk mengelola data taksonomi, seperti menambah data taksonomi, melihat, mengubah dan menghapus data taksonomi.

l. Modul Faq

Modul faq digunakan untuk mengelola data faq seperti menambah data faq, melihat, mengubah dan menghapus faq.

m. Modul Team

Modul team digunakan untuk mengelola data team, seperti menambah data kesesuaian iklim, melihat, mengubah dan menghapus data team.

n. Modul Tentang Jahe

Modul tentang jahe digunakan untuk mengelola data tentang jahe, seperti manfaat jahe, nama lain jahe dan kandungan kimia, admin dapat menambah data tentang jahe, melihat, mengubah dan menghapus data tentang jahe.

o. Modul User

Modul user digunakan untuk mengelola data user sebagai admin, seperti menambah data user, melihat, mengubah dan menghapus data user.

3.10. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non – fungsional merupakan analisis yang digunakan untuk mendefinisikan hal – hal yang berkaitan dengan sistem yang berjalan. Berikut adalah kebutuhan non fungsional sistem:

a. Ketepatan (*Correctness*)

Web service menyajikan data – data yang benar sesuai dengan data tentang tanaman jahe.

b. Integritas (*Integrity*)

Web service memiliki kemampuan untuk dapat mengawasi akses dari setiap pengguna terhadap data.

c. Keandalan (*Reliability*)

Web service dapat menjalankan fungsi dan menampilkan data sesuai yang telah ditentukan.

d. Uji Coba (*Testability*)

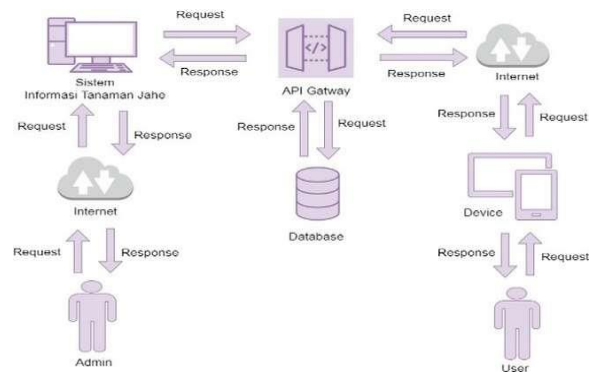
Web service dapat diuji coba untuk memastikan fungsi yang telah ditentukan pada web service.

e. Portabilitas (*Portability*)

Web service dapat diuji coba untuk memastikan fungsi yang telah ditentukan pada web service.

3.11. Analisis Arsitektur Sistem

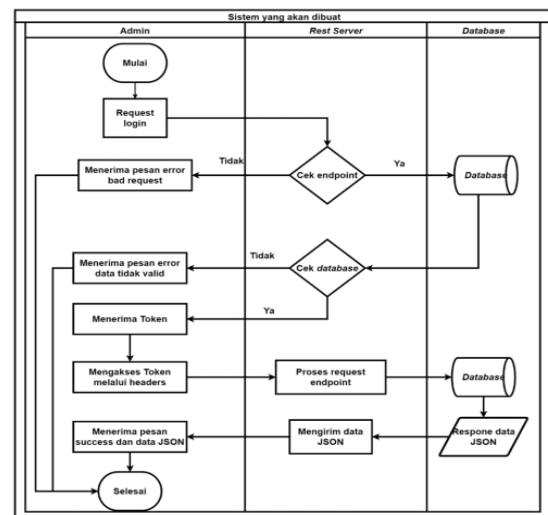
Analisis arsitektur sistem merupakan analisis yang digunakan untuk merancang arsitektur suatu *web service* sesuai dengan model *client-server*. *Web service* mengimplementasikan komunikasi dua arah, dimana client dapat mengirimkan request ke *server*, diproses dan ditampilkan sebagai respon. *Client* dan *server* tidak terhubung secara langsung, tetapi menggunakan file *web service* sebagai perantara. Di pencarian, format data yang digunakan adalah JSON. Dengan demikian, proses seleksi serta pengelolaan sistem manajemen saat mengakses *database* tidak akan dikelola langsung oleh *server* tetapi melalui layanan perantara *web service*. Arsitektur yang akan dibuat dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur Sistem

3.12. Analisis Sistem yang Akan Dibuat

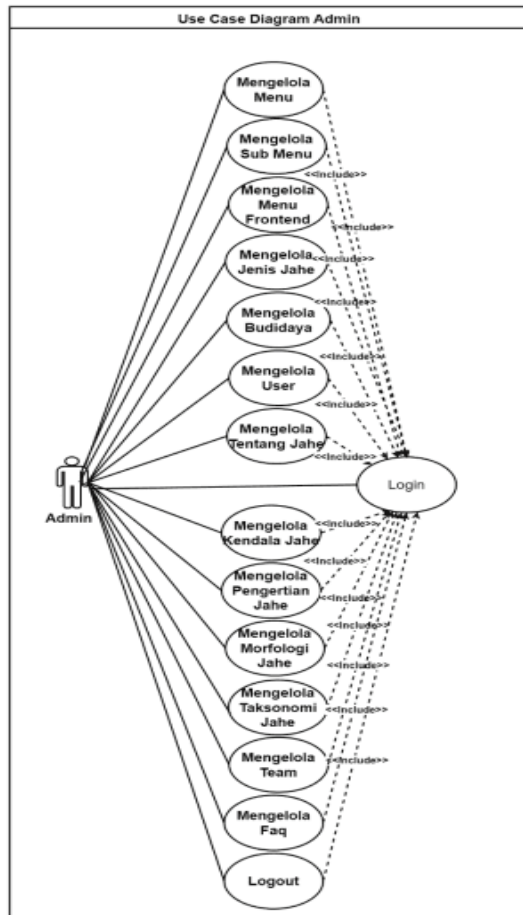
Analisis sistem yang dilakukan dengan menjelaskan bagaimana sistem atau alur proses kegiatan yang dibuat. Sistem yang akan diterapkan pada penelitian ini adalah untuk menyediakan *web service* yang akan digunakan oleh admin. Dapat dilihat pada Gambar 3.



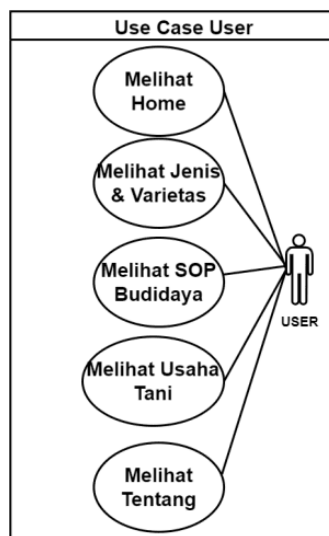
Gambar 3. Analisis Sistem yang Akan Dibuat

3.13. Perancang Sistem

Pada tahap ini dibuat sketsa desain dengan menggunakan bahasa pemodelan UML untuk menjelaskan struktur sistem melalui detail diagram *use case*, serta menyajikan detail desain *database* berdasarkan hasil analisis sebelumnya. Langkah ini juga mencakup perancangan implementasi *web service* dengan menerapkan pendekatan REST untuk memastikan komunikasi antarmuka yang efisien dan terukur. Keseluruhan proses desain ini melibatkan analisis ekstensif untuk memastikan kelengkapan fungsionalitas sistem, sebagai dasar untuk implementasi lebih lanjut yang direncanakan dalam gambar 4 dan 5.



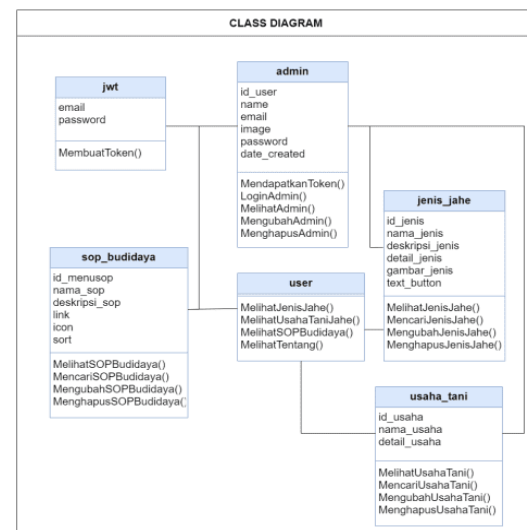
Gambar 4. Use Case Diagram Admin



Gambar 5. Use Case Diagram User

3.14. Class Diagram

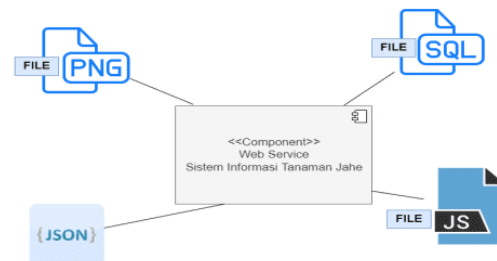
Class diagram diagram digunakan untuk menggambarkan himpunan kelas, antarmuka, kolaborasi dan relasi antara kelas serta antarmuka. Class diagram dapat dilihat pada Gambar 6.



Gambar 6. Class Diagram

3.15. Component Diagram

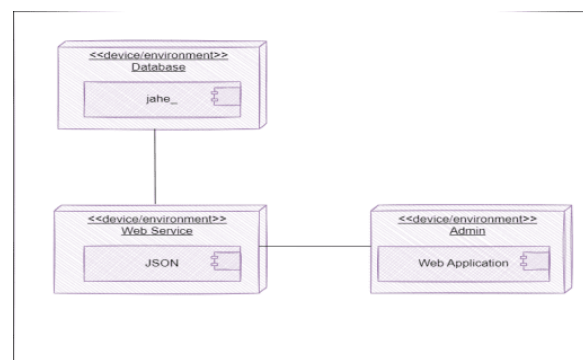
Component diagram menggambarkan struktur dan relasi antar komponen serta keterkaitan antara komponen lainnya dengan sistem. Component diagram dapat dilihat pada Gambar 7.



Gambar 7. Component Diagram

3.16. Deployment Diagram

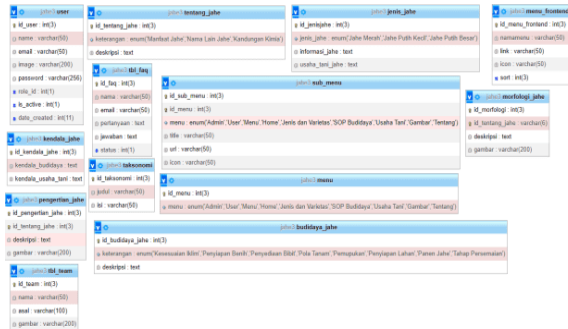
Deployment diagram menggambarkan hubungan antar komponen perangkat lunak dan komponen perangkat keras dan menampilkan konfigurasi dari masing – masing komponen diatas saat dijalankan. Deployment diagram dapat dilihat pada gambar 8.



Gambar 8. Deployment Diagram

3.17. Perancangan Database

Perancangan database untuk *web service* terdiri dari beberapa table. Perancangan *database* ini dimaksudkan untuk mendukung kebutuhan pengelolaan data dari sistem yang akan dibuat. *Database* ini digunakan untuk penyimpanan data, pengelolaan data berkaitan dengan data tanaman jahe, budidaya jahe, usaha tani jahe dan jenis jahe. Dapat dilihat pada Gambar 9.



Gambar 9. Database

4. HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi

Pada hasil implementasi *web service* yang sudah dibangun terdapat. 15 *web service*. Dalam aplikasi ini yaitu sebagai berikut:

4.2. Modul Registrasi Admin

Dokumentasi *service* registrasi admin yang berfungsi untuk membuat akun admin dapat dilihat pada Tabel 2.

Tabel 2. Registrasi Admin

Modul Registrasi Admin	
URL	localhost:3001/user/registrasi
Method	POST
Parameter	-
Body	name, email, image, password, date_created

4.3. Modul Login Admin

Dokumentasi *service login* admin berfungsi untuk mendapatkan token, token tersebut bisa dipakai untuk mengakses sistem dapat dilihat pada Tabel 3.

Tabel 3. Login Admin

Modul Login	
URL	localhost:3001/user/login
Method	GET
Parameter	-
Body	email, password

4.4. Modul Menu Frontend

Dokumentasi menu berfungsi untuk mengelola data menu. Dapat dilihat pada Tabel 4.

Tabel 4. Menu Frontend

URL	Method	Params	Deskripsi
/menu_frontend	GET	-	Untuk menampilkan semua data menu frontend
/menu_frontend/:id	GET	id_menu_frontend	Untuk menampilkan data menu frontend berdasarkan parameter tertentu
/menu_frontend	POST	id_menu_frontend, namamenu, link, icon, sort	Untuk menambah data menu frontend
/menu_frontend	PUT	id_menu_frontend, namamenu, link, icon, sort	Untuk mengubah data menu frontend
/menu_frontend	DELETE	id_menu_frontend	Untuk menghapus data menu frontend

4.5. Modul Faq

Dokumentasi Faq berfungsi untuk mengelola data Faq. Dapat dilihat pada Tabel 5.

Tabel 5. Faq

URL	Method	Params	Deskripsi
/faq	GET	-	Untuk menampilkan semua data faq
/faq/:id	GET	id_faq	Untuk menampilkan data faq berdasarkan parameter tertentu
/faq	POST	id_faq, nama, email, pertanyaan, jawaban, status	Untuk menambah data faq
/faq	PUT	id_faq, nama, email, pertanyaan, jawaban, status	Untuk mengubah data faq
/faq	DELETE	id_faq	Untuk menghapus data faq

4.6. Modul Team

Dokumentasi Team berfungsi untuk mengelola data team. Dapat dilihat pada Tabel 6.

Tabel 6. Team

URL	Method	Params	Deskripsi
/team	GET	-	Untuk menampilkan semua data team
/team/:id	GET	id_team	Untuk menampilkan data team berdasarkan parameter tertentu
/team	POST	id_faq, nama, email, pertanyaan, jawaban, status	Untuk menambah data faq

URL	Method	Params	Deskripsi
/team	PUT	id_faq, nama, email, pertanyaan, jawaban, status	Untuk mengubah data faq
/team	DELETE	id_team	Untuk menghapus data team

4.7. Pengujian

Pengujian dimaksudkan untuk mengamati hasil menjalankan melalui data uji dan untuk memverifikasi fungsionalitas perangkat lunak. Pengujian dalam

penelitian ini menggunakan metode *blackbox*. Metode *blackbox* melakukan pengecekan validasi terhadap hasil yang dikeluarkan oleh sistem pada saat sistem menerima perintah. Dapat dilihat pada Tabel 7.

Tabel 7. Pengujian

No	URL	Hasil Pengujian	Keterangan	Output yang diharapkan	Status
1	localhost:3001/user/register (POST)	{"status": 201, "success": true, "message": "Berhasil Tambah Data", "data": {}}	Pengujian endpoint untuk membuat akun admin	Response dari web service memberikan status 201 dan berhasil menambah data	Berhasil
2	localhost:3001/user/login (GET)	{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlZ2ODUzOTTA4NDR9.Fdf9SAXfQzFfl1cwZWkoY0EqMewDd_LumnUR7UOAzQI"}	Pengujian endpoint login untuk pengelolaan dan admi	Response dari web service meberikan token	Berhasil
3	localhost:3001/menu_frontend (GET)	{"status": 200, "success": true, "message": "Semua Data Menu Frontend", "data": {}}	Pengujian endpoint untuk menampilkan semua data menu frontend	Response dari web service memberikan status 200 dan menampilkan semua data menu frontend	Berhasil
4	localhost:3001/faq (GET)	{"status": 200, "success": true, "message": "Semua Data Faq", "data": {}}	Pengujian endpoint untuk menampilkan semua data faq	Response dari web service memberikan status 200 dan menampilkan semua data faq	Berhasil
5	localhost:3001/team (GET)	{"status": 200, "success": true, "message": "Semua Data Team", "data": {}}	Pengujian endpoint untuk menampilkan semua data team	Response dari web service memberikan status 200 dan menampilkan semua data team	Berhasil

5. KESIMPULAN DAN SARAN

Hasil kesimpulan penelitian yang sudah dilakukan adalah sudah dibangun sebuah web service untuk sistem informasi tanaman jahe dengan menggunakan metode REST API, maka dapat disimpulkan yaitu dapat memberikan informasi tentang tanaman jahe untuk diolah dan di tampilkan pada web service sistem informasi tanaman jahe. Penggunaan metode REST API berhasil diterapkan pada pembuatan web service untuk pertukaran data pada platform yang berbeda. Dan pada keamanan sudah memakai JSON Web Token (JWT) untuk pertukaran data antar platform agar lebih aman. sebuah web service yang disediakan untuk mengelola sistem informasi tanaman jahe dengan menggunakan framework Node JS dan Express JS sebagai teknologi untuk pengembangan web service. Pengujian untuk web service menggunakan metode black box dan menggunakan software postman dinyatakan berhasil.

Berdasarkan hasil penelitian yang sudah dilakukan, pengembangan web service dengan framework Node JS dan Express JS, diharapkan pengembangan selanjutnya dapat dilakukan untuk mengembangkan bagian mobile app ataupun dengan platform yang berbeda seperti iOS.

DAFTAR PUSTAKA

- [1] H. D. Setyaningrum dan C. Saparinto. *Jahe*. Jakarta: Penebar Swadaya, 2013.
- [2] K. Putri. M. *Khasiat dan Manfaat Jahe Merah*. Semarang: ALPRIN, 2009.
- [3] R. C. Fiecry. R. Alam. S. N. Rahmi. "Implementasi JSON Web Token pada Authentication dengan Algoritma HMAC SHA-256". *Jurnal Sistem Informasi (SISTEMASI)* 12, no. 1, 2023: 195.
- [4] Bahrudin. Wakkang H. Irianto B. "Implementasi Web Service Dengan Metode REST API Untuk Integrasi Data Covid 19 di Sulawesi Selatan". *Jurnal Sintaks Logika (JSilog)* 2, no.1, 2022.
- [5] Hutahean, Jeperson. *Konsep Sistem Informasi*. Yogyakarta: Deepublish, 2014.
- [6] S. Ramos. N. E. Michelle Tan. "Pengembangan Sistem Informasi Pelatihan Web Menggunakan Teknologi Web Service dan Framework Laravel". *Jurnal TECHNO Nusa Mandiri* 16, no. 1, 2023 : 52.
- [7] R. B. Prasetyo, P.T. Adhie. "Rancang Bangun Web Service API Aplikasi Sentralisasi Produk UMKM Pada UPTD Plut KUMKM Provisi Lampung". *Jurnal Teknologi dan Sistem Informasi (JTSI)* 02, no. 1, 2021: 60.
- [8] T. Davide, P. Claus, L. Valentina. "Architectural

- Patterns for Microservice: A Systematic Mapping Study". ResearchGate. 2018.
- [9] Jatnika, Hendra. *Pengantar Sistem Basis Data Memahami Konsep Dasar dan Tuntunan Praktis Perancangan Database*. Yogyakarta: CV. Andi Offset. 2013.
- [10] R. Alam, S. Heni, N. Rizal. "Keamanan RESTful Web Service Menggunakan Json Web Token(JWT) HMAC SHA-512". Jurnal Nasional Teknik Elektro dan Teknologi Informasi 07. no. 2. 2018: 132.
- [11] G. Rohmat, R. Alam. "Json Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service". Jurnal Edukasi dan Penelitian Informatika 05. no. 1. 2019.
- [12] R. S. Pressman, *Software Engineering: a Practitioner's Approach*, Fifth edition, 2010.