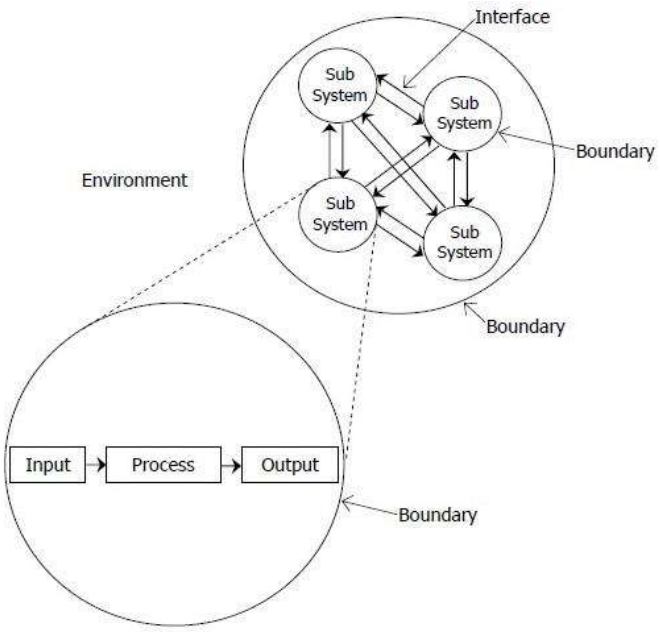
BAB II

TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem informasi merupakan komponen sistem yang saling keterkaitan untuk menghasilkan suatu informasi dalam bidang tertentu. Sistem informasi dikelompokkan untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan (Lorasponelsar, Zuhdi and Santoso, 2019). Sistem informasi dirancang untuk mentransformasikan data kedalambentuk informasi yang berguna (Ibrahim, Keni and Triyanita, 2021). Konsep karakteristik sistem dapat dikatakan sederhana apabila memiliki gambaran yang jelas dan terstruktur. Berikut merupakan penjelasan karakteristik dari sistem yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Karakteristik Sistem (Trimahardhika and Sutinah, 2017)

1. *Components*

Sistem terdiri dari sejumlah bahan yang saling berkolerasi dan membentuk satu kesamaan. Komponen sistem akan membentuk suatu subsistem yang memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu secara keseluruhan.

2. *Boundary*

Area sistem dapat membatasi antar sistem lainnya dengan lingkup luarnya. Penentu sistem akan memungkinkan sistem akan terpandang sebagai satu kesamaan yang tidak dapat terpisahkan.

3. *Environtment*

Aliran pada luar area sistem dapat mempengaruhi operasi sistem disebut dengan area luar sistem. Area luar sistem dapat bernilai dan merugikan sistem tersebut. Area tersebut memiliki nilai akan menjadi sebuah kekuatan bagi sistem yang terjaga dan terpelihara. Area luar yang merugikan harus terkendali, namun jika tidak terkendali akan mengganggu kelangsungan hidup sistem.

4. *Interface*

Sarana terhubung pada sistem dengan subsistem yang lain disebut dengan *interface*. *Interface* dapat memungkinkan sumber daya mengalir dan mendapatkan hasil menjadi masukan untuk subsistem yang lain dengan melewati penghubung dan membentuk satu kesamaan.

5. *Input*

Kekuatan yang terdapat pada sistem akan masuk dapat berupa *maintenance input* dan *signal input*. Sebagai contoh, pada unit sistem komputer

dan program adalah *maintenance input* yang dapat diaplikasikan pada komputer. Sementara data adalah sinyal *input* yang akan mendapatkan hasil menjadi sebuah informasi.

6. *Output*

Hasil yang telah dikerjakan dan dikelompokan akan menjadi sebuah *output* yang berguna. *Output* tersebut menjadi *input* bagi subsistem yang lain. Seperti contoh, *output* yang dihasilkan adalah informasi yang dapat digunakan sebagai *input* untuk pengambilan suatu keputusan bagi subsistem lainnya.

7. *Procces*

Sistem memiliki proses yang mengubah *input* menjadi *output*. Sebagai contoh, sistem akuntansi akan mengerjakan data transaksi menjadi sebuah laporan yang diperlukan oleh manajemen.

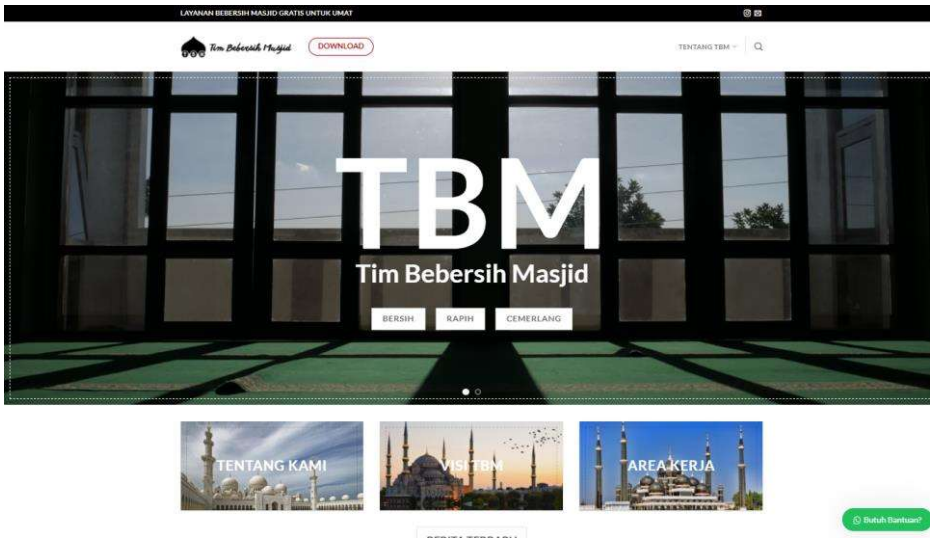
8. *Objective*

Sistem memiliki tujuan dan sasaran yang pasti dan sifat akan filosofi yang terjadi sesuai dengan sebab akibat yang menjadikan sebuah keharusan agar dapat terprediksi. Jika sistem tidak memiliki alur, maka operasi sistem tidak akan berjalan. Sistem dapat dikatakan berhasil bila sesuai dengan alur yang telah ditentukan.

2.2. Sistem Informasi Tim Bebersih Masjid

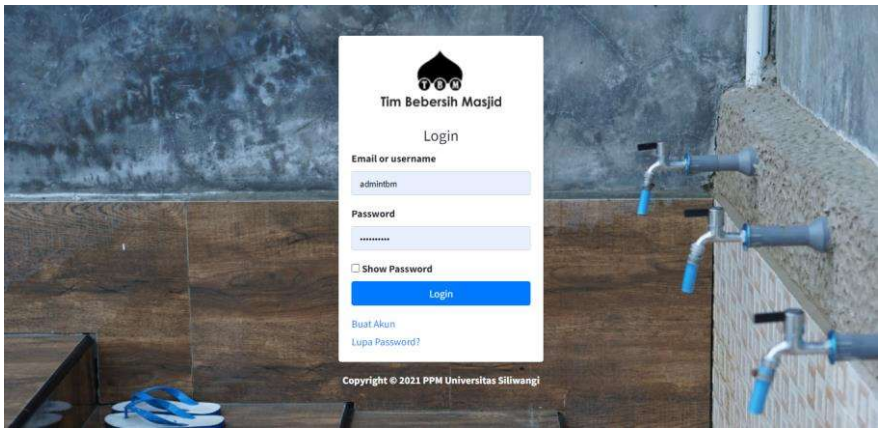
Beberapa yayasan sosial atau komunitas banyak melaksanakan bersih-bersih masjid. Setiap daerah sendiri memiliki kegiatan komunitas, lembaga, dan yayasan sosial tergerak untuk melaksanakan dalam hal memakmurkan masjid seperti bersih-

bersih masjid agar membantu pengurus masjid dalam hal kebersihan dan kenyamanan dari sebuah masjid. Sistem Informasi Tim Bebersih Masjid merupakan situs layanan masyarakat dengan fokus utmama untuk menjaga kebersihan masjid dan memakmurkan Masjid. Harapan dari *plaform* digital ini dapat mempermudah kinerja dalam menjaga kebersihan masjid dan berperan aktif dalam memakmurkan Masjid dengan informasi-informasi layanan keagaman yang dapat di akses dengan mudah.



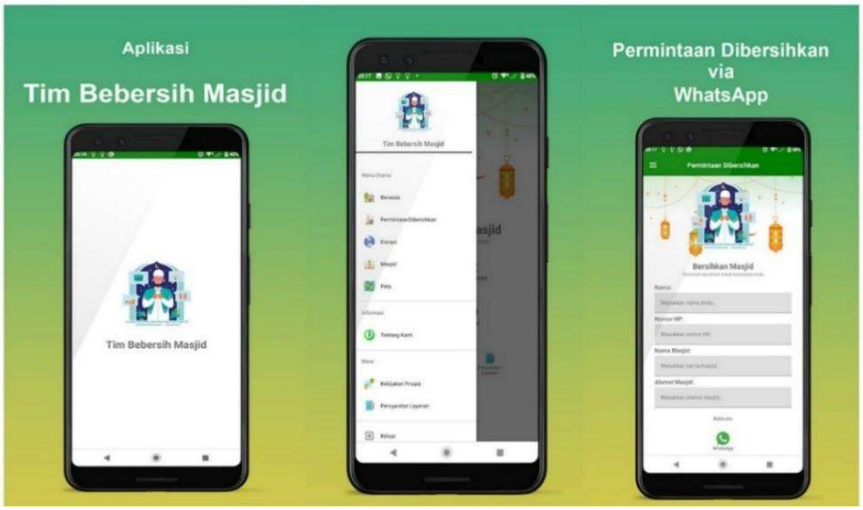
Gambar 2.2 Sistem Informasi Tim Bebersih Masjid Berbasis *Web*

Gambar 2.2 merupakan tampilan awal dari Sistem Informasi Tim Bebersih Masjid berbasis *website* yang dapat diakses pada https://masjidbersih.org/.



Gambar 2.3 Pendataan Digital Bersih-Bersih Masjid Untuk Petugas

Gambar 2.3 merupakan tampilan awal dari pendataan bersih-bersih masjid yang ada pada Sistem Informasi Tim Bebersih Masjid dan terinterasi pada aplikasi berbasis *mobile* yang dapat diakses https://app.masjidbersih.org/.



Gambar 2.4 Aplikasi Tim Bebersih Masjid Berbasis *Mobile Android*

Gambar 2.4 merupakan tampilan aplikasi Tim Bebersih Masjid berbasis *mobile android* yang dapat diunduh pada Google Play Store.

2.3. *Web Service*

Web service merupakan suatu fasilitas yang dapat menyediakan akses layanan dalam bentuk informasi atau data yang valid kepada sistem lain hingga berinteraksi dengan sistem tersebut dengan beberapa layanan yang tersedia (Zaman, 2017). Format pertukaran data dibuat dapat dipanggil dan diakses oleh aplikasi lain melalui internet. *Web service* melakukan penyimpanan data informasi dalam format *JSON* atau *XML*, sehingga data akan diakses oleh sistem lain meskipun berbeda *platform*, sistem operasi, dan bahasa pemrograman (Bunyamin and Syazili, 2019).

2.4. RESTful API

REST merupakan metode standar arsitektur komunikasi berbasis web yang digunakan untuk pengembangan *web service* berbasis *web* (Ehsan *et al.*, 2022). Implementasi arsitektur *REST* melalui *HTTP* (*Hypertext Transfer Protocol*) dan saat membaca halaman *web* berisi file *XML* atau *JSON* (Adi Pranata, Hijriani and Junaidi, 2018). *Application Programming Interface* (API) akan menentukan aturan untuk berkomunikasi dengan sistem *software* lainnya (Herfandi, Julkarnain and Hanif, 2022). *RESTful API* merupakan *interface* yang digunakan oleh dua (2) sistem komputer untuk bertukar informasi secara aman melalui internet (Adi Pranata, Hijriani and Junaidi, 2018).

2.5. JSON

JSON atau kepanjangan dari *JavaScript Object Notation* merupakan format data yang bekerja untuk pertukaran dan penyimpanan data. Penggunaan JSON

dapat bertukar data antar aplikasi sesuai dengan format standar. Selain itu, contoh penggunaan JSON sebagai format untuk bertukar data client dan server atau antar aplikasi, yaitu format pertukaran data RESTful API (Warsito, Ananda and Triyanjaya, 2017).

2.6. *JSON Web Token* (JWT)

JSON Web Token (JWT) merupakan format pengamanan informasi pribadi menjadi *encode* dalam bentuk JSON agar membentuk *payload* dari *JSON Web Signature* (JWS). Permintaan oken biasanya ditandatangani secara digital atau diamankan secara kriptografis, sebagai contoh *Message Authentication Code* (MAC) atau dapat dienkripsikan (Jánoky, Ekler and Levendovszky, 2021).

2.7. *Keyed-Hash Message Authentication Code* (HMAC)

Keyed-Hash Message Authentication Code (HMAC) adalah algoritma untuk menghitung nilai *Message Authentication Code* (MAC) yang dikombinasikan dengan sebuah fungsi *hash* dan kunci rahasia (Ichwan, Gustian and Nurjaman, 2018). HMAC ini digunakan untuk memeriksa integritas data dan otentikasi dari sebuah pesan yang dikirimkan. Fungsi yang dapat diperoleh dalam algoritma HMAC adalah untuk memakai fungsi *hash*, untuk mengatur kunci *private*, untuk analisa kriptografi, dan mempermudah kinerja agar keamanan lebih baik (Ramadhani, Ramadhani and Basit, 2020).

2.8. *Secure Hash Algorithm* 256 (SHA-256)

Secure Hash Algorithm 256 (SHA-256) merupakan fungsi hash yang sering digunakan hingga saat ini dan belum ada yang dapat memecahkan algoritma fungsi *hash* SHA-256 (Sulastri and Putri, 2018). Algoritma SHA-256 mempunyai delapan langkah pengerjaan, yaitu tambahkan *bit padding*, panjang *append*, *parsing* pesan, inisialisasi nilai *hash*, mempersiapkan jadwal pesan, inisialisasi delapan variabel kerja a, b, c, d, e, f, g, dan h dengan nilai *hash* (i-1), menjumlahkan hasil akhir a, b, c, d, e, f, g, h dengan inisial *hash value* $H^{(i)}$, dan *output* (Saputra and Nasution, 2019).

2.9. Postman

Postman merupakan aplikasi yang berfungsi sebagai REST CLIENT untuk pengujian REST API. Postman digunakan oleh para *developer* pembuat API sebagai *tools* dalam menguji API yang telah dibuat. Dokumentasi API dibuat lengkap dengan memanfaatkan Postman dalam mempermudah proses pengembangan projek, karena setiap *developer* memiliki acuan yang akurat dalam penggunaan API (Galindra Wardhana, Arwani and Rahayudi, 2020).

2.10. Penelitian Terkait (*State of The Art*)

Penelitian terkait akan menjawab pertanyaan yang berhubungan pada permasalahan skalabilitas dan kinerja JWT sebagai teknologi pendukung dalam penerapan arsitektur *web service*. Penelitian mengenai JWT, algoritma SHA-256, dan *Rational Unified Process* disajikan pada tabel 2.1 *state of the art*.

Tabel 2.1 *State of The Art*

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
1.	SHA-*512 Algorithm on JSON Web Token for Restful Web Service-Based Authentication*	(Rasyada, 2022)	JWT, Algoritma SHA-512	Token pada algoritma SHA-512 memiliki nilai *hash* yang lebih besar dibandingkan token dengan algoritma SHA-256. Pembuktian ukuran token yang lebih besar, lebih panjangnya algoritma SHA-512 dibandingkan SHA-256.
2.	Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order	(Edy *et al.*, 2021)	RESTful API, JWT, Algoritma SHA-256	Penerapan RESTful API pada aplikasi sales order berjalan baik berbasis *web* maupun *mobile* menjadi lebih mudah. Penggunaan autentifikasi JWT pada RESTful API menjadi aplikasi lebih aman karena aplikasi tidak dapat diakses jika tidak menggunakan token.
3.	*JSON Web Token Implementation for Dynamic Access Rights Authentication in* Klinik Pratama UPN "Veteran" Yogyakarta *Application Based on RESTful*	(Danutirta *et al.*, 2021)	RESTful API, JWT, Algoritma SHA-256	Implementasi autentikasi dengan *JSON Web Token* akan meningkatkan keamanan data. Pengujian tersebut dapat diimplementasikan yang berisikan data *username* dan *password* yang dienkripsi hingga didekripsi menjadi data parameter untuk otorisasi sistem.

Tabel 2.2 *State of The Art* (Lanjutan 1)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
4.	Implementasi *JSON Web Token Authentication* pada Aplikasi Pembayaran Berbasis *Mobile*	(Sopingi, Nastiti and Majid, 2021)	JWT, REST API	Penelitian terkait sistem pembayaran pada *mobile application* akan memproses data pembayaran semester. Implementasi *Web Service* dan *JSON Web Token Authentication* akan memberikan pengamanan data dengan menunjukan bahwa pada aplikasi pembayaran yang dibangun *problem* pada aplikasi yang sesuai dengan kebutuhan fungsional yang diharapkan.
5.	Implementasi Fitur Keamanan dengan *JSON Web Token* dan Fitur *Geo-Tagging* pada Aplikasi *Web Service Training From Home*	(Hibsy and Wibowo, 2020)	JWT, *Geo-Tagging*, REST API	Hasil implementasi *web service* pada *mobile application* dengan JWT dan *geo-tagging* telah bekerja dengan baik dan akurat. Aspek keamanan hak akses saat proses *transfer* data mendapatkan token dengan jangka waktu 24 jam. Sistem otentikasi bila ada kesalahan maka akses token tidak akan berhasil digunakan. Aplikasi yang dibangun memberikan presentase baik.

Tabel 2.3 *State of The Art* (Lanjutan 2)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
6.	Perancangan Bot Untuk Monitoring *Server* Dari Serangan *Distributed Denial Of Service* dan Implementasi JSON Web Token Pada Sistem Notifikasi Serangan	(Rahman, Seta and Astriratma, 2020)	JWT, Algoritma HMAC, *Bot Token*	Pengujian tersebut bekerja dengan cara masuknya serangan DDOS berjenis UDP Flooding dan SYN Flooding pada *web server* yang diamankan oleh bot. Alat tersebut akan mengukur waktu dari serangan DDOS hingga muncul pesan pada *stopwatch*.
7.	*Web Service Security System Analysis With Rest Architecture Using The Aes Method With* JWT	(Saputra and Setianto, 2020)	SOAP, Metode AES, JWT	Keamanan *web service* menjadi hal yang sangat penting, karena setiap layanan yang bekerja menjadi lebih aman dan tidak sembarang *user* dapat mengakses layanan tersebut. Token yang pada JWT membuat *user* memiliki otentikasi untuk mengakses layanan yang tersedia dan enkripsi tambahan menggunakan metode AES agar data tersebut terjaga dengan baik.

Tabel 2.4 *State of The Art* (Lanjutan 3)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
8.	Implementasi *Web Service* pada Perusahaan Logistik menggunakan *JSON Web Token* dan Algoritma Kriptografi RC4	(Mochammad Rizky Royani and Wibowo, 2020)	JWT, Algoritma Kriptografi RC4	Pengujian metode dan analisa program mendapatkan hasil implementasi, yaitu membuktikan jika perusahaan lebih efisien dalam proses penginputan transaksi yang dapat diukur dari sisi waktu pemrosesan yang rata-rata hanya 176 ms. Sedangkan sebelum adanya aplikasi tersebut memproses transaksi selama 20 menit untuk satu proses layanan.
9.	Implementasi JSON Web Token Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita	(Setiawan and Purnamasari, 2020)	JWT, Algoritma SHA-512	Pengujian yang dilakukan pada penelitian ini, yaitu melakukan implementasi *JSON Web Token* (JWT) dengan algoritma SHA-512 pada aplikasi BatikKita untuk mempercepat proses otentikasi hingga meningkatkan keamanan saat otentikasi.

Tabel 2.5 *State of The Art* (Lanjutan 4)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
10.	*API Features Individualizing of Web Services: REST and SOAP*	(Soni and Ranga, 2019)	REST API, SOAP, *Web Service*	*Soap* memiliki kekuatan yang lebih berat dibandingkan *Rest* dan membutuhkan lebih banyak waktu CPU. REST lebih mudah dipahami dan dikembangkan, ringan, bekerja dengan format apa pun saat SOAP adalah terbatas pada XML saja.
11.	Pengembangan Aplikasi Evaluasi Dosen Berbasis Android dengan Keamanan *JSON Web Token* (JWT)	(Junirianto, 2019)	JWT, REST API, *Waterfall*	Hasil implementasi dan pengujian aplikasi dapat terintegrasi pada sistem aplikasi evaluasi dosen berbasis Android. Sistem tersebut menggunakan keamanan *JSON Web Token* pada *web service* untuk membantu proses pertukaran data lebih terjamin keamanannya.

Tabel 2.6 *State of The Art* (Lanjutan 5)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
12.	Analisa Algoritma SHA-256 Untuk Mendeteksi Orisinalitas Citra Digital	(Saputra and Nasution, 2019)	Kriptografi, Algoritma SHA-256	Hasil pengujian pada algoritma SHA-256 dengan penerapan teknologi kriptografi akan mendeteksi perubahan pada citra hasil pemindaian ijazah dan transkrip nilai walaupun hanya satu piksel yang berubah. Perbedaan tersebut akan menghasilkan nilai *hash* yang signifikan.
13.	Sistem Pembayaran Uang Kuliah Terintegrasi (Studi Sekolah Tinggi Teknologi (STT) Dumai Dengan Bank Syariah Mandiri)	(Suhaidi and Nugraha, 2018)	JWT, REST API, Algoritma SHA-256	Membuat sebuah sistem informasi dan integrasi data menjadi langkah untuk mengatasi permasalahan keterbatasan informasi. Langkah tersebut dibuktikan dengan pembuatan arsitektur integrasi, sistem yang aman, dan ketersediaan arsitektur integrasi sistem untuk pengamanan data yang berkesinambungan.

Tabel 2.7 *State of The Art* (Lanjutan 6)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
14.	Perancangan *Application Programming Interface* (API) Berbasis *Web* Menggunakan Gaya Arsitektur *Representational State Transfer* (*Rest*) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit	(Adi Pranata, Hijriani and Junaidi, 2018)	JWT, REST API	Penelitian REST mendapatkan hasil implementasi pada sistem administrasi pasien klinik perawatan kulit. REST API menggunakan *path*, *query* dan request method dengan klasifikasi *resource* yang digunakan. Pengujian otentikasi REST API mendapatkan hasil JWT yang bekerja pada sistem *back-end server*.
15.	Aplikasi Pengamanan Data Dengan Algoritma Kriptografi AES-256 Berbasis REST API	(Mahdhani and Siswanto, 2018)	REST API, Algoritma Kriptografi AES-256	Pengujian pada REST API untuk pengamanan data dengan algortima kriptografi AES 256 tidak dapat dibaca oleh pihak yang tidak berhak bagi yang tidak memiliki *access token*. Proses tersebut akan bergantung pada koneksi internet saat pengaksesan, sehingga proses kirim dan *request* akan cepat atau lambat.

Tabel 2.8 *State of The Art* (Lanjutan 7)

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
16.	RESTFul *Web Service* Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ	(Tanaem, Manongga and Irian, 2016)	RESTful API, JWT, Algoritma SHA-256	Penelitian ini mendapatkan hasil arsitektur RESTFul API pada *Web Service* yang aman bagi PT. XYZ. RESTFul API pada *Web Service* dibangun menggunakan JWT dengan mengamankan sebuah komunikasi akan atau telah terjadi. Pengamanan komunikasi yang terjadi memungkinkan untuk mengintegrasikan sumber daya dari PT. XYZ dengan menggunakan aplikasi yang berbeda.

2.11. Matriks Penelitian

Matrik penelitian merupakan perbandingan antara penelitian sebelumnya dengan penelitian yang akan dilakukan. Indikator untuk melakukan sebuah matriks penelitian, yaitu dari berbagai sumber jurnal yang telah dikaitkan pada *state of the art*. Beberapa jurnal terkait berhubungan dengan penggunaan arsitektur komunikasi, teknologi, tujuan dan objek penelitian dengan penelitian yang sedang dilakukan. Tabel 2.2 menggambarkan perbedaan penelelitian yang diusulkan dengan penelitian-penelitian terkait.

Tabel 2.9 Matriks Penelitian

No.	Penulis dan Tahun	Parameter								
		JWT	**Citra Digital**	**HMAC SHA-256**	**HMAC SHA-512**	**Pemindaian Citra**	***Respons Time***	***Size Data***	***Blackbox***	**Pengujian Data Tokem**
1.	Analisa Algoritma SHA-256 Untuk Mendeteksi Orisinalitas Citra Digital (Saputra and Nasution, 2019)	-	√	√	-	√	-	-	-	-
2.	Pengembangan Aplikasi Evaluasi Dosen Berbasis Android dengan Keamanan *JSON Web Token* (Junirianto, 2019)	√	-	√	-	-	-	-	√	-
3.	Implementasi Fitur Keamanan dengan *JSON Web Token* dan Fitur *Geo-Tagging* pada Aplikasi *Web Service Training From Home*	√	-	√	-	-	√	-	√	-

	(Hibsy and Wibowo, 2020)									
4.	*JSON Web Token Implementation for Dynamic Access Rights Authentication in* Klinik Pratama UPN "Veteran" Yogyakarta *Application Based on RESTful* (Danutirta *et al.*, 2021)	√	-	√	-	-	-	-	-	√
5.	SHA-*512 Algorithm on JSON Web Token for Restful Web Service-Based Authentication* (Rasyada, 2022)	√	-	-	√	-	√	√	-	-
6.	Penelitian Usulan (Ficry, 2022)	√	-	√	-	-	√	√	-	√

Berdasarkan pada tabel 2.2 bahwa matriks penelitian ini memiliki kelebihan, kekurangan, dan perbandingan. Pengambilan indikator matriks penelitian ini dikaitkan pada *state of the art* yang telah dicantumkan pada tabel 2.2.

Penelitian (Saputra and Nasution, 2019) berfokus pada penerapan keamanan informasi pada citra digital untuk mendeteksi orisinalitas citra. Hasil pemindaian ijazah dan transkrip nilai menjadi kelebihan pada penelitian ini dengan menerapkan pemindaian citra gambar dengan menggunakan HMAC SHA-256. Namun, kelemahan pada penelitian ini hanya berfokus pada pengujian berdasarkan parameter pemindaian citra dan menghasilkan perbedaan pada nilai *hash*. Parameter yang digunakan pada penelitian ini, yaitu penerapan citra digital berdasarkan pemindaian citra dan menggunakan HMAC SHA-256.

Penelitian (Junirianto, 2019) berfokus pada penerapan *web service* untuk teknologi JWT. Penelitian ini membangun sebuah aplikasi berbasis *mobile* yang menerapkan teknologi JWT sebagai keamanan *authentication*. Kelebihan pada

penelitian ini, yaitu dapat mengintegrasikan sistem aplikasi evaluasi dosen berbasis *android*. Kekurangan pada penelitian ini hanya berfokus pada pengembangan aplikasi berbasis *mobile* yang hanya menerapkan pengujian menggunakan *blackbox* dan *user interface* yang diterapkan pada penelitian ini belum dibuat lebih menarik hingga penambahan fitur-fitur sesuai dengan kebutuhan *user* pada lingkungan Politeknik Pertanian Negeri Samarinda. Parameter yang digunakan pada penelitian ini, yaitu menerapkan teknologi JWT dengan menggunakan algoritma HMAC SHA-256 dan pengujian menggunakan *blackbox*.

Penelitian (Hibsy and Wibowo, 2020) berfokus pada implementasi fitur keamanan dengan JWT pada aplikasi presensi *training from Home* berbasis *mobile*. Kelebihan pada penelitian ini selain memfokuskan penerapan JWT, penambahan fitur *geo-tagging* menjadi keunggulan pada penelitian ini untuk menentukan posisi saat melakukan presensi *training*. Selain memiliki kelebihan atau keunggulan pada penelitian ini, ada kekurangan pada penelitian ini yang tidak melakukan pengujian aplikasi pada fitur *geo-tagging* dari segi kinerjanya. Terlihat bahwa pengujian dari segi kinerja sistem hanya berfokus pada penerapan JWT saja. Parameter yang diterapkan pada penelitian ini, yaitu teknologi JWT, *geo-tagging*, dan menggunakan algoritma HMAC SHA-256.

Penelitian (Danutirta *et al.*, 2021) berfokus pada implementasi *authentication* menggunakan JWT. Kelebihan dari penelitian ini, yaitu melakukan pengujian data token yang tidak diterapkan pada tiga (3) penelitian yang telah dicantumkan pada matriks penelitian. Pengujian data token ini menjadi tolak ukur implementasi *authentication* menggunakan JWT dikarenakan implementasi ini telah

diimplementasikan yang terenkripsi dan menjadi parameter untuk otorisasi sistem sesuai dengan hak akses penggunanya. Selain pengujian data token, penerapan *signature* untuk memverifikasi pengirim JWT berasal dari sumber yang benar dan untuk memastikan bahwa pesan tidak berubah. Kekurangan dari penelitian ini, yaitu penerapan JSON Web Token hanya terdapat pada *user login* dikarenakan konsep *default account* dalam pembangunan aplikasi Klinik Pratama UPN "Veteran" Yogyakarta, yang mana tidak ada autentifikasi token disisi backend sehingga proses autentikasi hanya berjalan disisi *frontend*. Parameter yang digunakan pada penelitian ini menerapkan teknologi JWT dengan algoritma HMAC SHA-256 dan pengujian data token.

Penelitian (Rasyada, 2022) berfokus pada implementasi keamanan data dengan teknologi JWT. Sama seperti penelitian (Junirianto, 2019), namun kelebihan dari penelitian ini berfokus pada performa kinerja aplikasi dan penerapan algoritma HMAC SHA-512. Namun, kelemahan pada penelitian ini memfokuskan pengujian algoritma dengan satu (1) sistem operasi yang menghasilkan performa kinerja yang tidak stabil. Terlihat dari pengujian *respons time* pada testing ke-20 yang menghasilkan 1516 ms dan *size data* yang dihasilkan berjumlah 3,09 kb pada algoritma HMAC SHA-256. Pengujian pada algoritma HMAC SHA-512 pada testing ke-20 menghasilkan 1511 ms pada *respons time* dan 3,26 kb hasil *size data*. Parameter yang diterapkan pada penelitian ini, yaitu menerapkan teknologi JWT dengan HMAC SHA-512 dan pengujian performa kinerja JWT.

Kesimpulan yang dapat diambil dan menjadi penelitian usulan (Ficry, 2022), yaitu meimplementasikan JWT dengan HMAC SHA-256. Implementasi pada

penelitian ini memfokuskan pengamanan *authentication* saat melakukan akses data yang diterapkan menggunakan REST API. Parameter yang diterapkan pada penelitian ini berfokus pada performa kinerja dari JWT dengan algoritma HMAC SHA-256 berdasarkan *respons time* dan *size data* yang dihasilkan saat pengujian menggunakan sistem operasi Windows 7 (*virtual machine*).