

## Implementasi JSON Web Token Authentication pada Aplikasi Pembayaran Berbasis Mobile

Sopingi<sup>2</sup>, Faulinda Ely Nastiti<sup>2</sup>, Arvian Sofyan Majid<sup>3</sup>

<sup>1</sup>Sistem Informasi, Universitas Duta Bangsa Surakarta

<sup>2</sup>Sistem Informasi, Universitas Suta Bangsa Surakarta

<sup>3</sup>Teknik Informatika, Universitas Duta Bangsa Surakarta

E-mail: sopingi@udb.ac.id

### Abstrak

*Alat pembayaran saat ini semakin berkembang dari pembayaran tunai menjadi pembayaran non tunai. Seperti pembayaran yang dilakukan di Universitas Duta Bangsa yang sebelumnya dilakukan dengan melakukan setor tunai langsung pada teller bank, kemudian saat ini dikembangkan ke pembayaran non tunai melalui aplikasi pembayaran berbasis mobile. Mahasiswa dapat melakukan pembayaran dari berbagai chanel pembayaran dan secara otomatis ditransaksikan di sistem informasi keuangan. Akan tetapi terdapat permasalahan keamanan data mahasiswa yang disimpan di dalam aplikasi dan proses authentication pada setiap proses transaksi data antara aplikasi dengan web service sistem keuangan. Penulis memberikan solusi berupa penggunaan JSON Web Token (JWT) pada proses authentication dan penerapan hash token pada setiap data yang dikirim ke server untuk memvalidasi keaslian data berasal dari aplikasi pembayaran berbasis mobile. Dalam penelitian ini penulis menggunakan metode Rapid Application Development (RAD) dan melakukan pengujian teknik User Acceptance Testing (UAT) kepada pengguna. Berdasarkan hasil User Acceptance Testing diketahui bahwa aplikasi pembayaran berbasis mobile mampu melakukan authentication menggunakan JWT dan berhasil mengamankan data dengan penggunaan hash token pada setiap request ke web service.*

**Kata Kunci:** *Pembayaran, Aplikasi Mobile, JWT, RAD, UAT.*

### Abstract

*Payment instruments are currently increasing from cash payments to non-cash payments. As was done at Duta Bangsa University, which was previously done by making cash deposits directly at bank tellers, now it is being developed into non-cash payments through mobile-based payment applications. Students can make payments from various payment channels and are automatically transacted in the financial information system. However, there are problems with student security data stored in the application and the authentication process in every data transaction process between the application and the financial web service system. The author provides a solution in the form of using a JSON Web Token (JWT) in the authentication process and applying a hash token to any data sent to the server to validate the authenticity of data originating from mobile-based payment applications. In this study, the authors use the Rapid Application Development (RAD) method and test the User Acceptance Testing (UAT) technique to users. Results Based on User Acceptance Testing, it is known that mobile-based payment applications are able to authenticate using JWT and successfully generate data using hash tokens on every request to a web service..*

**Keywords:** *Payment, Mobile App, JWT, RAD, UAT.*

## 1. Pendahuluan

Perubahan digitalisasi berkembang secara cepat di Indonesia, demikian pula potensinya di masa depan. Tren digitalisasi tersebut mempengaruhi sendi-sendi perekonomian, mengubah pola transaksi masyarakat, baik perorangan atau institusi dan merubah kebiasaan konvensional termasuk di sektor keuangan (*Perkembangan Sistem Pembayaran Di Indonesia*, n.d.). Lembaga pendidikan salah satu sektor yang ikut terdampak sehingga perlu mengikuti perubahan digitalisasi terutama di sistem keuangan.

Universitas Duta Bangsa (UDB) dalam hal ini menjadi objek penelitian yang dilakukan penulis, selama ini UDB sudah memiliki sistem pembayaran tetapi terdapat kendala dalam keamanan data transaksi mahasiswa yang disimpan di dalam aplikasi dan proses authentication pada setiap proses transaksi data antara aplikasi dengan web service sistem keuangan, kendala yang ditemukan berupa kegagalan transaksi yang dilakukan oleh mahasiswa padahal uang sudah berhasil terkirim ke rekening UDB, kendala yang lain adalah pembayaran berhasil tetapi terjadi kesalahan transaksi sehingga harus dilakukan perbaikan transaksi secara manual. Setelah dilakukan evaluasi terhadap aplikasi pembayaran ditemukan celah keamanan yang mempengaruhi perubahan data.

Penulis memberikan solusi berupa penggunaan JSON Web Token (JWT) pada proses authentication dan penerapan hash token pada setiap data yang dikirim ke server untuk memvalidasi keaslian data berasal dari aplikasi pembayaran berbasis mobile. JWT adalah objek JSON sebagai cara aman untuk transaksi informasi antara dua sistem. JWT menyimpan data yang terdiri dari header, payload, dan tanda tangan (Madden, 2020). Hash adalah algoritma untuk mengubah data. Data yang dimasukkan dikonversi menjadi angka, huruf, atau simbol lain menjadi karakter terenkripsi tanpa mengubah ukuran. Data yang terenkripsi lewat fungsi hash tidak bisa dikembalikan (Aumasson, 2017).

## 2. Metodologi

Metode pengembangan sistem yang digunakan oleh penulis dalam penelitian yaitu menggunakan metode RAD (*Rapid Application Development*). Tujuan dari penggunaan metode sistem development yaitu memberikan sistem aplikasi yang dapat menyelesaikan permasalahan dari pengguna, tetapi banyak dari pengembang tidak melibatkan para pengguna sistem secara langsung. Dampaknya adalah sistem yang dibuat jauh dari tujuan pengguna, sehingga pengguna enggan untuk menggunakan sistem yang dibuat, bahkan akan menolak untuk menggunakannya.

Pada saat RAD diimplementasikan, maka pengguna dapat menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan. Penggunaan metode RAD dapat menghasilkan sebuah sistem secara cepat karena yang dikembangkan dapat menyelesaikan kebutuhan dari pengguna sehingga meminimalisir waktu untuk pengembangan ulang setelah menyelesaikan tahap implementasi (Irmayani, 2019). Tahapan-tahapan dari metode RAD antara lain:

- a. Rencana Kebutuhan (*Requirement Planning*)
  - 1) Pengguna dan peneliti mengidentifikasi tujuan dari dibangunnya aplikasi pembayaran administrasi,
  - 2) Berorientasi pada pemecahan permasalahan.
- b. Proses Desain (*Design Workshop*)
  - 1) Fase desain dan penyempurnaan,
  - 2) Menggunakan pendukung keputusan sistem untuk membantu proses penyetujuan dari pengguna,
  - 3) Developer dan penganalisis membangun dan menunjukkan gambaran visual desain dan alur kerja kepada pengguna,
  - 4) Pengguna memberikan tanggapan dari *prototype* yang sudah dibuat oleh *developer*,
  - 5) Menyempurnakan *prototype* yang dirancang berdasarkan tanggapan dari pengguna.

c. Implementasi (*Implementation*)

- 1) Setelah sistem baru dibangun, maka dilanjutkan proses pengujian dan diperkenalkan kepada para pengguna, yaitu mahasiswa dan bidang administrasi,
- 2) Ketika proses membuat sistem baru, maka tidak diperlukan untuk menjalankan sistem lama secara parallel.

**3. Hasil dan Pembahasan**

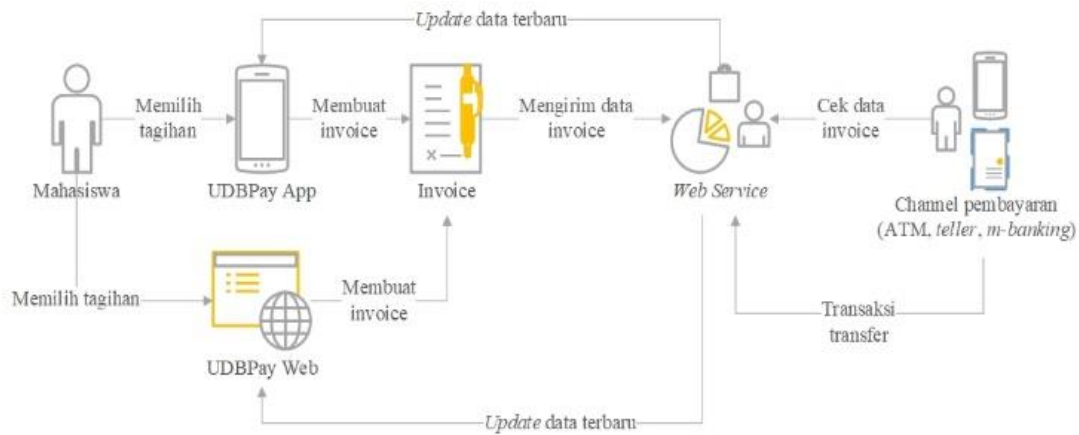
**3.1 Perencanaan Kebutuhan**

Peneliti dan pengguna melakukan diskusi dan analisis terhadap permasalahan dan kebutuhan sistem. Adapun kebutuhan yang diperlukan pada sistem agar dapat berjalan dengan baik meliputi:

- a. Sistem mampu melakukan *authentication* mahasiswa dengan menggunakan JWT.
- b. Sistem menampilkan beberapa tagihan program studi dan tagihan lainnya
- c. Sistem dapat memberikan pilihan kepada mahasiswa mengenai pembayaran penuh atau mengangsur
- d. Sistem dapat melakukan tambah tagihan sebagai daftar tagihan *invoice*
- e. Sistem dapat membuat *invoice*
- f. Sistem dapat membatalkan *invoice*
- g. Sistem menampilkan total tagihan yang sudah terbayar dan total kekurangan.

**3.2 Proses Desain**

Rancangan aplikasi pembayaran dengan proses komputerisasi dapat dilihat pada gambar 1 sebagai berikut .



Gambar 1. *Workflow* Aplikasi Pembayaran

Pada aplikasi pembayaran berbasis mobile penulis melakukan pengelolaan dengan menggunakan *Web Service* dengan *JWT authentication*. *Web service* mampu melakukan transmisi data dengan tingkat *performance* pada *response time* kurang dari 1 detik (Sopingi et al., 2021). Struktur dari *Web Service* dari setiap layanan aplikasi pembayaran meliputi:

- a. Login Mahasiswa. *Web service* akan melakukan proses mengirim data dari aplikasi menuju *server* dengan menggunakan method *POST*. Adapun proses *request* dan *response* dari *web service* dapat dilihat pada tabel 1 berikut:

Tabel 1. Login *Web Service*

<i>Request</i>	<i>Response</i>	<i>Method</i>
{ "username":String(9), "password":String(6)	{ "data": { "no_pend": String(9),	<i>POST</i>

<i>Request</i>	<i>Response</i>	<i>Method</i>
} }	"nim": String(9), "nama": String(30) "kode prodi": String(5), "tahun angkatan": String(4) "id_jns_daftar": int(2), "nm_jenj_didik": String(2), "nm lemb": String(30) "gelombang": String(15), "gel": String(12), "email": String(30), "fakultas": int(2), "nama_fakultas": String(6) "divisi": String(1), "username": String(9), "nomor pembayaran": String(12), "api_bank": String(10) "prefix_bank": String(6) }, "pesan": String(20), "rc": String(6), "token": String(100) }	

- b. Tagihan. Untuk mendapatkan daftar tagihan dilakukan proses dengan method GET pada *web service*. Method GET berfungsi sebagai metode untuk mengambil data dari *server* kemudian mengirim data tersebut pada aplikasi. Adapun proses *request* dan *response* dengan menggunakan *web service* dapat dilihat pada tabel 2 berikut:

Tabel 2 Tagihan *Web Service*

<i>Request</i>	<i>Response</i>	<i>Method</i>
Header JWT Authorization: Bearer <token>	{ "rc": String(6), "pesan": String(100), "data": [ { "id_tagihan": String(5) "id_biaya": int(4), "kategori biaya": String(10), "nama biaya": String(100), "tagihan jumlah": int(7), "tagihan terbayar": int(7), "tagihan potongan": int(7), "tagihan kekurangan": int(7) }, ] }	GET

- c. Invoice. Dibutuhkan proses mengambil data dari server guna mendapatkan informasi yang ada pada halaman *invoice*. Untuk mengambil data tersebut dibutuhkan sebuah *method* GET pada *web service* agar informasi *invoice* dapat dikirim atau ditampilkan pada aplikasi. Adapun *request* dan *response* yang dibutuhkan pada tabel 3 berikut:

Tabel 3. Invoice *Web Service*

<i>Request</i>	<i>Response</i>	<i>Method</i>
Header JWT Authorization: Bearer <token>	{ "rc": String(3), "msg": String(20), "id": String(30), "nomorPembayaran": String(12), "idPelanggan": String(9), "nama": String(50), "idTagihan": String(10), "totalNominal": String(6), "apiBank": String(20),	GET

<i>Request</i>	<i>Response</i>	<i>Method</i>
	<pre>"prefixBank": String(6), "createdAt": String(40), "informasi": [ { "label_key": String(30), "label_value": String(50) }, { "label_key": String(30), "label_value": String(50) }, { "label_key": String(30), "label_value": String(50) } ], "rincian": [ { "kode rincian": String(100) "nominal": String(6) } ] }</pre>	

- d. *Generate Invoice*. Sedangkan pada proses buat *invoice* dilakukan dengan menggunakan *method* POST, karena pada proses tersebut akan dilakukan pengiriman data dari aplikasi menuju ke server. Adapun *request* dan *response* yang dibutuhkan dalam proses mengirim data seperti pada tabel 4 berikut:

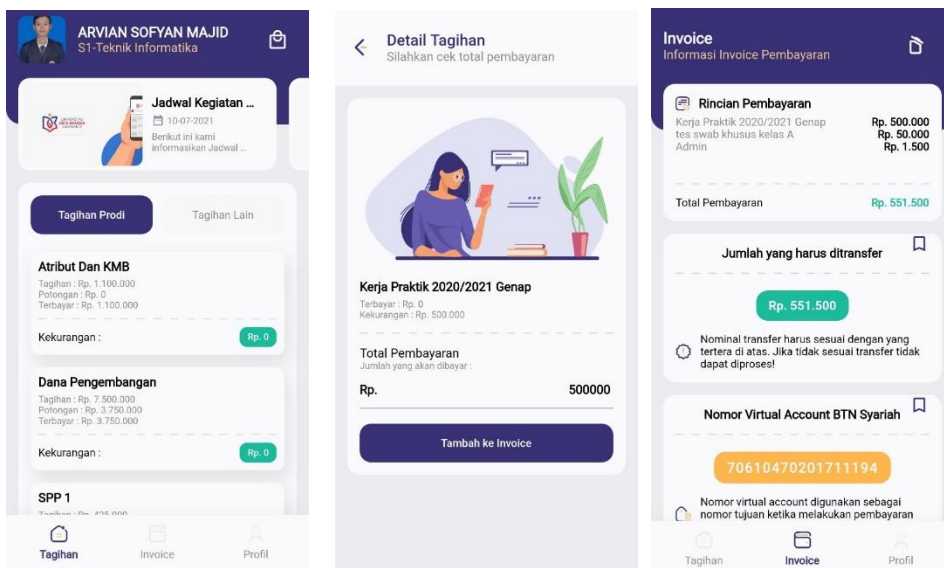
Tabel 4. *Generate Invoice Web Service*

<i>Request</i>	<i>Response</i>	<i>Method</i>
<pre>{ "nomorPembayaran":String(12), "no pend": String(10), "nim": String(9), "nama": String(50), "fakultas": String(8), "prodi": String(20), "total nominal": int(8), "api bank": String(10), "billing_detail": [ { "jenis_biaya": String(30), "id tagihan": String(10), "nama biaya": String(50), "jumlah": int(8), "keterangan": String(30), } ] }</pre>	<pre>{ "data": { "id": String(30), "id_tagihan": String(10), "NomorPembayaran": String(12), "no_pend": String(10), "nim": String(9), "nama": String(50), "fakultas": String(8), "prodi": String(20), "total_nominal": int(8), "api_bank": String(10), "billing detail": [ { "id": String(9), "id_billing": String(9), "jenis_biaya": String(30), "id_tagihan": String(10), "nama biaya": String(50), "tagihan": int(8), "potongan": int(8), "jumlah": int(8), "kekurangan": int(8), "keterangan": String(30), }, { "id": String(9), "id_billing": String(9), "jenis_biaya": String(30), "id_tagihan": String(10), "nama biaya": String(50), "tagihan": int(8), "potongan": int(8), "jumlah": int(8), "kekurangan": int(8), "keterangan": String(30), } ] } }</pre>	POST

Request	Response	Method
	<pre> }, ] },  "pesan": String(20), "rc": String(3), "tanggal": String(30), }                     </pre>	

### 3.3 Implementasi

Implementasi aplikasi pembayaran pada sisi *client* dengan menggunakan *Flutter* dan di *deploy* ke platform android dan ios. Berikut adalah tampilan aplikasi pembayaran berbasis *mobile*.



Gambar 2. Proses Membuat *Invoice*

Pada halaman *invoice* ditampilkan nomor *virtual account* yang digunakan sebagai nomor tujuan transaksi pembayaran melalui ATM, M-Banking atau teller. Adapun tampilan proses *inquiry* pada pembayaran seperti pada gambar 3 berikut:



Gambar 3. Proses *Inquiry* melalui Teller

Penulis melakukan pengujian dengan menggunakan teknik *User Acceptance Testing (UAT)*. *User acceptance testing* merupakan proses pengujian yang dilakukan oleh pengguna sistem. Hasil dari pengujian dapat dijadikan bukti bahwa sistem dapat membantu para pengguna (Hambling & Goethem, 2013). Berikut hasil pengujian yang dilakukan.

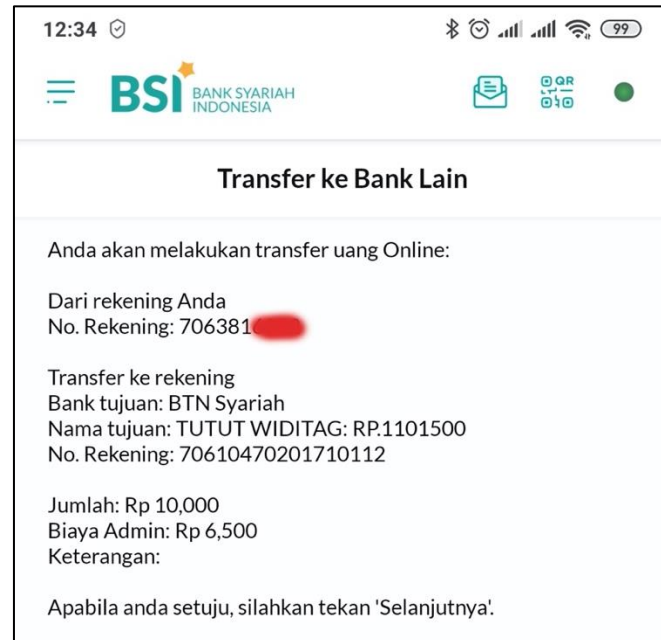
Tabel 5. Hasil Pengujian Tambah Daftar Invoice

Data Normal			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Melakukan tambah daftar invoice	Menampilkan halaman pembayaran	Tampil halaman pembayaran beserta total pembayaran	Berhasil
Klik "Tambah ke Invoice"	Menampilkan pesan pemberitahuan "Tagihan berhasil ditambahkan"	Menampilkan pesan pemberitahuan "Tagihan berhasil ditambahkan"	Berhasil
Hitung total pembayaran	Tampil hasil total pembayaran	Menampilkan hasil total pembayaran	Berhasil
Data Salah			
<i>Mengosongkan text field total nominal</i>	Akan menampilkan pesan pemberitahuan "Silahkan isi nominal pembayaran"	Menampilkan pesan pemberitahuan dan <i>user</i> tidak dapat melanjutkan ke proses selanjutnya	Berhasil
Mengisi total nominal lebih besar dari nominal tagihan	Tampil pesan pemberitahuan "Jumlah nominal tidak boleh lebih"	Menampilkan pesan pemberitahuan dan <i>user</i> tidak dapat melanjutkan ke proses selanjutnya	Berhasil
<i>Mengosongkan text field total nominal</i>	Akan menampilkan pesan pemberitahuan "Silahkan isi nominal pembayaran"	Menampilkan pesan pemberitahuan dan <i>user</i> tidak dapat melanjutkan ke proses selanjutnya	Berhasil

Tabel 6. Hasil Pengujian Generate Invoice

Data Normal			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik "Buat Invoice"	Menampilkan halaman invoice	Tampil halaman invoice dan informasi terkait proses transaksi	Berhasil
Memanggil data nomor <i>virtual account</i>	Menampilkan data nomor <i>virtual account</i> tiap <i>user</i>	Menampilkan data nomor <i>virtual account</i> tiap <i>user</i>	Berhasil
Tambah biaya admin	Tampil nominal biaya admin	Menampilkan nominal biaya admin dan menghitung total pembayaran	Berhasil
Klik "Salin Nomor"	Menampilkan pesan pemberitahuan "Nomor sudah tersalin" dan menyalin nomor <i>virtual account</i>	Menampilkan pesan pemberitahuan "Nomor sudah tersalin" dan <i>user</i> dapat melakukan <i>copy paste</i>	Berhasil
Data Salah			
Mengosongkan daftar invoice pembayaran	Tombol "Buat Invoice" berubah menjadi "Buat Daftar Tagihan"	Tombol "Buat Invoice" berubah menjadi "Buat Daftar Tagihan" dan <i>user</i> tidak dapat melanjutkan proses transaksi	Berhasil

Pengujian juga dilakukan untuk proses pembayaran dengan menggunakan m-banking seperti gambar 4 berikut ini.



Gambar 4. Proses Pembayaran menggunakan M-Banking

#### 4. Kesimpulan dan Saran

##### 4.1 Kesimpulan

Dari pembahasan yang telah dilakukan, adapun beberapa hal yang dapat disimpulkan dari sistem pembayaran berbasis mobile, meliputi :

- Sistem pembayaran berbasis mobile memproses data meliputi detail tagihan, *invoice*, profil mahasiswa dan riwayat pembayaran.
- Implementasi *Web Service* dan *JWT Authentication* mampu memberikan pengamanan data dan kemudahan akses ke *database server*.
- Hasil dari pengujian yang dilakukan menunjukkan bahwa pada aplikasi pembayaran yang dibangun terbebas dari *bug* dan sesuai dengan kebutuhan fungsional yang diharapkan.

##### 4.2 Saran

Dari penelitian yang sudah dilakukan maka penulis memberikan saran untuk penelitian berikutnya dapat dilakukan pembahasan terkait pembayaran berbasis mobile yang terintegrasi dengan berbagai chanel pembayaran melalui QRIS dan menambahkan enkripsi pada data yang dikirim ataupun diterima

##### Daftar Pustaka

- Aumasson, A.-P. (2017). *Serious Cryptography A Practical Introduction to Modern Encryption*. No Starch Press.
- Hambling, B., & Goethem, P. van. (2013). *User Acceptance Testing a Step-By-Step Guide*.
- Irmayani, D. (2019). Rekayasa Perangkat Lunak. In *Jurnal Informatika* (Vol. 2, Issue 3). Informatika Bandung. <https://doi.org/10.36987/informatika.v2i3.201>
- Madden, N. (2020). *API Security in Action*. Manning.
- Perkembangan Sistem Pembayaran di Indonesia*. (n.d.). <https://www.bi.go.id/id/fungsi-utama/sistem-pembayaran/default.aspx#floating-2>
- Sopingi, Ety Meikati, & Wijiyanto. (2021). Application of the Rational Unified Process Method

in Web Service Development Payment System Integration with Multibank Virtual Accounts. *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, 5(1), 75–88. <https://doi.org/10.37339/e-komtek.v5i1.565>