

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

Selama masa menjalani kerja magang pada departemen Pengembangan TI di PT Tujju Bio Indonesia / TUJJU Media, dilakukan dengan menjabat sebagai *back end developer intern*, dimana memiliki tanggung jawab dalam mengerjakan *back-end*, dan *database* website yang diarahkan oleh supervisi, Bapak Dimas Habib, selaku *Chief Technology Officer* (CTO).

Aplikasi yang dikembangkan merupakan salah satu proyek yang memang sudah direncanakan untuk dibangun demi melengkapi fitur *Tujju Management System* (TMS). Oleh karena itu, setiap minggunya dilakukan presentasi kepada supervisi dan senior untuk dilakukan pengulasan untuk memenuhi kebutuhan perusahaan.

#### 3.2 Tugas yang Dilakukan

Selama melaksanakan pekerjaan magang, tugas utama yang diberikan PT.Tujju Bio Indonesia adalah untuk merancang dan membangun sebuah *Representational State Transfer Application Programming Interface* (REST API) yang merupakan sebuah API *Electronic Wallet*(E-Wallet) yang memiliki fungsi sebagai tabungan pribadi, sharing tabungan, dan *Payment Gateway* untuk setiap *Vendor*, *Biller*, dan *multifinance* yang menjadi mitra. Proyek *Representational State Transfer Application Programming Interface* (REST API) diberi nama *Digital Pocket API*, proyek ini nantinya akan diimplementasikan kedalam seluruh product PT. Tujju Bio Indonesia sebagai sistem pembayaran.

Dalam proyek ini tugas yang diberikan oleh supervisi adalah membuat rancangan *database*, *entity relationship diagram*, serta membangun *controller* dan *end-point*. Tugas pertama yang dilakukan dalam proyek ini adalah membuat *entity relationship diagram* (ERD) sesuai dengan kebutuhan yang telah diberikan oleh pembimbing lapangan yakni Bapak Dimas Habib, berikutnya setelah rancangan ERD diselesaikan rancangan tersebut kemudian dievaluasi oleh Bapak Farizal Hamami apakah rancangan tersebut sudah sesuai dengan yang diinginkan.

Selanjutnya membuat rancangan database sesuai dengan database yang telah dievaluasi, setelah selesai pembuatan database. Dilanjutkan dengan pembu-

atan *controller* dan *endpoint* yang akan menerima permintaan HTTP Request dan mengirim *response* kembali kepada *client*, setelah selesai tugas selanjutnya untuk melengkapi pekerjaan adalah melakukan pengujian pada endpoint tersebut menggunakan aplikasi Postman untuk mengecek dan membuat dokumentasi dari *query*, *form input*, serta *response* endpoint API yang telah dibangun.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang dimulai pada 2 Agustus 2021, diawali dengan pengenalan tentang perusahaan Tuju Media, penyambutan, pengenalan lingkungan kerja, dan pembelajaran mandiri tentang GitHub/GitLab sebagai project repository. Selanjutnya pelaksanaan kerja magang dapat dibagi menjadi 4 tahapan, yaitu *knowledge test*, *mockup project*, *company standard code generalization*, *main project*.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang

Minggu Ke -	Pekerjaan yang dilakukan
1	<i>Knowledge test</i> membuat website buku tamu menggunakan framework laravel.
2	<i>Knowledge test</i> melanjutkan pembuatan website buku tamu dengan fitur baru seperti, <i>file storage</i> , <i>query builder</i> , <i>eager loading</i> , <i>error handling</i> .
3	<i>Knowledge test</i> melanjutkan pembuatan website buku tamu dengan fitur baru seperti, <i>collection</i> , <i>middleware</i> , <i>serialization</i> .
4	<i>Knowledge test</i> melanjutkan pembuatan website buku tamu dengan fitur baru seperti, <i>HTTP Client</i> , <i>API Resources</i> , <i>Mail</i> , <i>Queue</i> , <i>Export</i> , <i>Cache</i> , <i>Console</i> .
5	<i>Mockup project</i> persiapan dan pemberian tugas oleh supervisi, pembuatan website <i>Point of sales</i>
6	<i>Mockup project</i> perancangan <i>database</i> dan <i>flowchart</i> , serta inisialisasi proyek kedalam repository dan melakukan pembuatan otentikasi untuk API, <i>middleware</i> yang akan digunakan dalam proyek.
7	<i>Mockup project</i> pembangunan <i>controller</i> , <i>endpoint</i> , <i>database migration</i> , <i>implementation to front-end</i> , <i>API Resources</i> dan melakukan dokumentasi kode. Untuk proyek <i>Point of sales</i> (POS).

Tabel 3.2. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
8	<i>Mockup project</i> pembangunan sistem <i>login</i> dan <i>register</i> menggunakan API, <i>Create, Read, Update, and Delete</i> (CRUD) untuk <i>employee</i> dan <i>product</i> .
9	<i>Mockup project</i> melakukan presentasi proyek kepada supervisi untuk dilihat apakah hasilnya sudah sesuai dan apa saja yang perlu direvisi. Melakukan revisi pada proyek serta melakukan penambahan fitur yang diinginkan POS. Menyelesaikan proyek POS.
10	<i>company standard code generalization</i> melakukan persiapan untuk proyek utama dengan mempelajari standarisasi kode yang terdapat pada perusahaan Tujju Media.
11	<i>Main project</i> melakukan pengarahan untuk project utama <i>Digital Pocket API</i> , pembuatan ERD, pembuatan migrasi <i>database</i> , pembuatan <i>login and register API</i> .
12	<i>Main project</i> pembangunan fungsi <i>Create, Read, Update, And Delete</i> (CRUD) untuk fitur transaksi. Pembuatan <i>payment gateway</i> untuk vendor, merchant, biller.
13	<i>Main project</i> mempresentasikan hasil yang telah dikerjakan untuk dilakukan pengulasan dan perlu untuk dilakukan revisi. Serta melakukan <i>brainstorming</i> dengan rekan - rekan IT.
14	<i>Main Project</i> merevisi <i>bug</i> yang terdapat aplikasi yang telah dibuat, melakukan beberapa penyesuaian agar program dapat berjalan dengan baik.
15	<i>Main project</i> pembuatan error handling, dokumentasi kode, dokumentasi <i>API endpoint</i> , serta memfinalisasi proyek yang telah dibuat.

Uraian magang dibagi menjadi 4 tahapan, yaitu perencanaan, perancangan, implementasi, dan pengujian. Tahapan tersebut digunakan untuk mengerjakan proyek Digital Pocket API pada PT. Tujju Bio Indonesia.

### 3.3.1 Perencanaan

#### A. Analisis Kebutuhan

Pada tahap ini diperlukan diskusi dengan supervisi mengenai seperti apa fitur dari Digital Pocket API, alur kerja dari aplikasi Digital Pocket API, dan data - data apa saja yang di perlukan untuk aplikasi Digital Pocket API. Aplikasi Digital Pocket API merupakan sebuah aplikasi dompet elektronik yang dapat digunakan untuk melakukan pembayaran pada setiap product - product / layanan yang disediakan oleh PT. Tujju Bio Indonesia.

Setelah melakukan diskusi ditemukan bahwa fitur - fitur yang diinginkan untuk terdapat pada aplikasi Digital Pocket API yaitu, *login, register, email verification, reset password, show balance, top up balance, create sub pocket, send balance, request payment, dan move balance to sub pocket*. Pembuatan aplikasi harus mengikuti ketentuan yaitu dibuat menggunakan bahasa pemrograman PHP dengan *framework Laravel* sesuai standarisasi kode yang telah diterapkan dalam setiap aplikasi yang dibangun oleh PT.Tujju Bio Indonesia.

#### B. Perangkat Lunak dan Perangkat Keras

Perangkat lunak yang digunakan dalam proses pengerjaan project Digital Pocket API adalah sebagai berikut:

1. Microsoft Visio digunakan dalam perancangan ERD.
2. PostgreSQL digunakan untuk memenejemen database lokal.
3. GitLab digunakan untuk repository proyek.
4. Visual Studio Code aplikasi editor pemrograman yang digunakan untuk mengerjakan proyek.
5. Postman digunakan untuk mengambil data API secara manual.

Perangkat keras yang digunakan dalam proses pengerjaan project Digital Pocket API adalah sebagai berikut:

1. Sistem Operasi : Microsoft Windows 10 Home

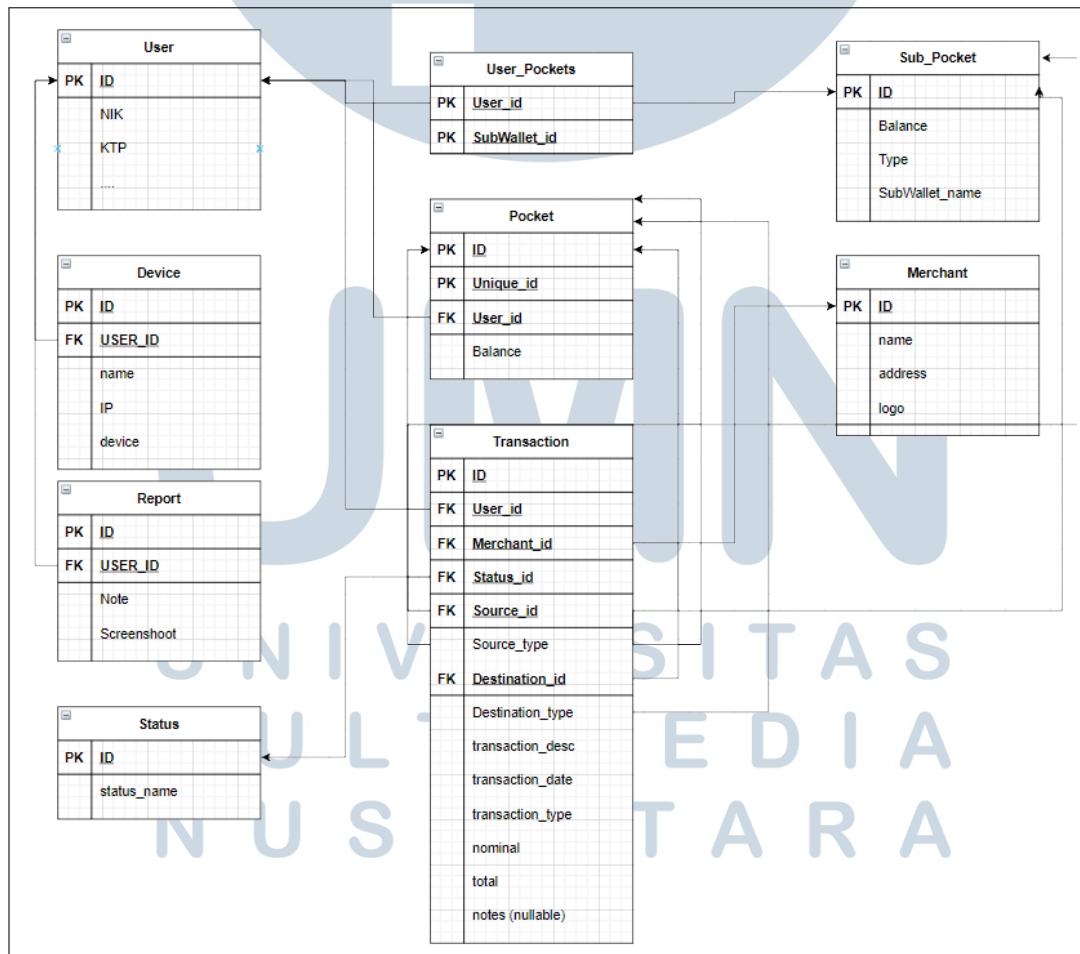
2. Processor : Intel(R) Core(TM) i5-9300H
3. Memori : 8 GB RAM
4. SSD : 512 GB
5. Graphic : NVIDIA Graphic Card Geforce RTX 2060

### 3.3.2 Perancangan

Tahap perancangan dimulai dengan membuat *Entity Relationship Diagram*(ERD), perancangan kebutuhan data, dan perancangan parameter.

#### A. *Entity Relationship Diagram*(ERD)

*Entity relationship diagram*(ERD) yang telah dirancang dalam pembangunan aplikasi Digital Pocket API adalah sebagai gambar yang terdapat pada Gambar3.1.



Gambar 3.1. ERD aplikasi back-end Digita Pocket API

Pada ERD yang telah dirancang terdapat total sejumlah 9 table yang akan gunakan dalam pembuatan aplikasi yaitu, *User*; *Device*, *Report*, *Status*, *UserPockets*, *Pocket*, *Transaction*, *SubPocket*, *Merchant*. Relationship dari table - table tersebut dapat dilihat pada gambar diatas untuk memudahkan dalam mengetahui hubungan dari setiap data yang ada.

## B. Perancangan Kebutuhan Data

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
enc_id	character varying	50		<input type="checkbox"/> No	<input type="checkbox"/> No
name	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
email	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
email_verified_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
password	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
nik	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
ktp	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
phone	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
remember_token	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.2. Rancangan Kebutuhan Data Table User

Table user berisikan data - data pribadi pengguna mulai dari nama, email, password, nik, dan nomor telepon. Table ini nantinya digunakan untuk transaksi data pengguna misalnya ketika pengguna melakukan *login*, *register*, atau melakukan *report*.



Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
user_id	bigint			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
status_id	bigint			<input type="checkbox"/> No	<input type="checkbox"/> No
source_id	bigint			<input type="checkbox"/> No	<input type="checkbox"/> No
destination_id	bigint			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
source_type	character varying	191		<input type="checkbox"/> No	<input type="checkbox"/> No
destination_type	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
transaction_desc	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
transaction_date	timestamp without time zone			<input type="checkbox"/> No	<input type="checkbox"/> No
transaction_type	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
nominal	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
total	numeric	8	0	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
notes	character varying	191		<input type="checkbox"/> No	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.3. Rancangan Kebutuhan Data Table Transaction

Table transaction berisikan data - data transaksi yang dilakukan menggunakan aplikasi mulai dari pembayaran, pengisian dana, dan perpindahan dana. Table ini nantinya digunakan untuk mencatat tanggal transaksi, pengguna yang melakukan transaksi, jumlah transaksi, serta tanggal terjadinya transaksi.

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
status_name	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.4. Rancangan Kebutuhan Data Table Status

Table status hanya menampung data informasi status transaksi apakah transaksi tersebut diterima, ditolak, ataupun pending.

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
unique_id	character varying	191		<input type="checkbox"/> No	<input type="checkbox"/> No
balance	numeric	15	0	<input type="checkbox"/> No	<input type="checkbox"/> No
user_id	bigint			<input type="checkbox"/> No	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.5. Rancangan Kebutuhan Data Table Pocket

Table pocket merupakan dompet yang dimiliki user di dalamnya terdapat informasi berupa jumlah dana, id unik, dan id pemilik dompet. Table pocket ini akan sering digunakan ketika terjadinya transaksi.

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
name	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
address	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
logo	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.6. Rancangan Kebutuhan Data Table Merchant

Table merchant merupakan dompet yang dimiliki user di dalamnya terdapat informasi berupa jumlah dana, id unik, dan id pemilik dompet. Table pocket ini akan sering digunakan ketika terjadinya transaksi.



Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
name	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
ip_address	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
device	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
user_id	bigint			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
status	boolean			<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.7. Rancangan Kebutuhan Data Table Device

Table device mendata pengguna masuk menggunakan perangkat apa dan dari ip mana, hal ini berguna untuk meningkat keamanan dari aplikasi untuk mengetahui apakah yang login merupakan orang yang berbeda.

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
balance	numeric	8	2	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
type	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
wallet_name	character varying	191		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
created_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone	0		<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.8. Rancangan Kebutuhan Data Table Sub Pocket

Table subpocket berisikan data wallet yang bukan merupakan wallet utama yang telah dibuat oleh pengguna sesuai dengan yang diinginkan masing - masing pengguna, misalnya sebagai tabungan, untuk dibagikan misalnya, dsb.

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
user_id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
sub_pocket_id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
created_at	timestamp without time zone		0	<input type="checkbox"/> No	<input type="checkbox"/> No
updated_at	timestamp without time zone		0	<input type="checkbox"/> No	<input type="checkbox"/> No

Gambar 3.9. Rancangan Kebutuhan Data Table User Pocket

Table ini hanya berisikan informasi mengenai siapa saja yang dapat mengakses wallet yang bukan merupakan wallet utama (*Sub Pocket*) / sebagai pivot untuk mengetahui akses *Sub Pocket*.

### C. Perancangan Parameter

KEY	VALUE
transaction_desc	Top Up Balance
transaction_type	TOPUP
nominal	10000000
total	10000000

Gambar 3.10. Rancangan Parameter Topup

Untuk tipe transaksi topup dibutuhkan beberapa parameter yaitu, nominal, total, transactiontype, transactiondesc. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

KEY	VALUE	DESCRIPTION
source_id	3	
source_type	pockets	
destination_id	8c5162438d8baea83549824ddedd44d2	Target Unique ID
transaction_desc	Jajan Bulanan, Bulan October	
transaction_type	SEND	
nominal	20000	
total	20000	

Gambar 3.11. Rancangan Parameter Send

Untuk tipe transaksi send dibutuhkan beberapa parameter yaitu, sourceid, sourcetype, destinationid, destinationtype, transactiondesc, transactiontype, nominal, total. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi. destinationid di gunakan untuk mendapatkan dompet yang akan dituju, destinationid ini menggunakan kode unik yang terdapat pada setiap dompet.

KEY	VALUE	DESCRIPTION
source_id	2	
source_type	pockets	
destination_id	6c69f2e2cb21a674273d03de8b0a0c8e	Target Unique ID
destination_type	pockets	
transaction_desc	Minta Uang Jajan Mak	
transaction_type	REQUEST	
nominal	20000	
total	20000	

Gambar 3.12. Rancangan Parameter Request

Untuk tipe transaksi request dibutuhkan beberapa parameter yaitu, sourceid, sourcetype, destinationid, destinationtype, transactiondesc, transactiontype, nominal, total. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi. destinationtype digunakan untuk mencari tahu tipe dompet apa yang dituju dompet utama ataupun dompet cadangan.

KEY	VALUE	DESCRIPTION
source_id	2	
source_type	pockets	
destination_id	1	Merchant ID
destination_type	merchant	
transaction_desc	Bayar Bakso Di Kang Bankso	
transaction_type	PAY	
nominal	20000	
total	20000	

Gambar 3.13. Rancangan Parameter Pay

Untuk tipe transaksi pay dibutuhkan beberapa parameter yaitu, sourceid, sourcetype, destinationid, destinationtype, transactiondesc, transactiontype, nominal, total. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi. destinationtype diisikan *merchant* untuk memberi tahu bahwa transaksi yang dilakukan ialah pembayaran kepada *merchant*.

KEY	VALUE	DESCRIPTION
destination_id	1	Id Pocket or Subpocket
destination_type	subpockets	Type Pockets or Subpocket
source_id	1	Id Pocket or Subpocket
source_type	pockets	Type Pockets or Subpocket
transaction_desc	Minta Uang Jajan Mak	
transaction_type	MOVE	
nominal	20000	
total	20000	

Gambar 3.14. Rancangan Parameter Move

Untuk tipe transaksi move dibutuhkan beberapa parameter yaitu, destinationid, destinationtype, sourceid, sourcetype, transactiondesc, transactiontype, nominal, total. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi. sourcetype digunakan untuk mengetahui dari sumber manakah dana didapatkan.

KEY	VALUE
transaction_id	38
source_id	2
source_type	pockets
transaction_desc	Maap Cuma Blsa Kasih Segini
transaction_type	SEND
nominal	20000
total	20000

Gambar 3.15. Rancangan Parameter Accept Request

Untuk tipe transaksi accept request dibutuhkan beberapa parameter yaitu, transactionid, sourceid, sourcetype, transactiondesc, transactiontype, nominal, total. Kebutuhan data lainnya akan didapatkan melalui *Bearer Token* untuk mengetahui data siapa yang melakukan transaksi. transactionid digunakan untuk mendapatkan data permintaan kepada pengguna.

KEY	VALUE
wallet_name	Tabungan Bersama
type	Shared

Gambar 3.16. Rancangan Parameter Create Sub Pocket

Parameter ini diperlukan untuk pembuatan dompet cadangan nama dompet, dan tipe dompet apakah dapat di sharing atau tidak.

KEY	VALUE
target_id	41
subpocket_id	Text ▾ 21

Gambar 3.17. Rancangan Parameter Link Sub Pocket

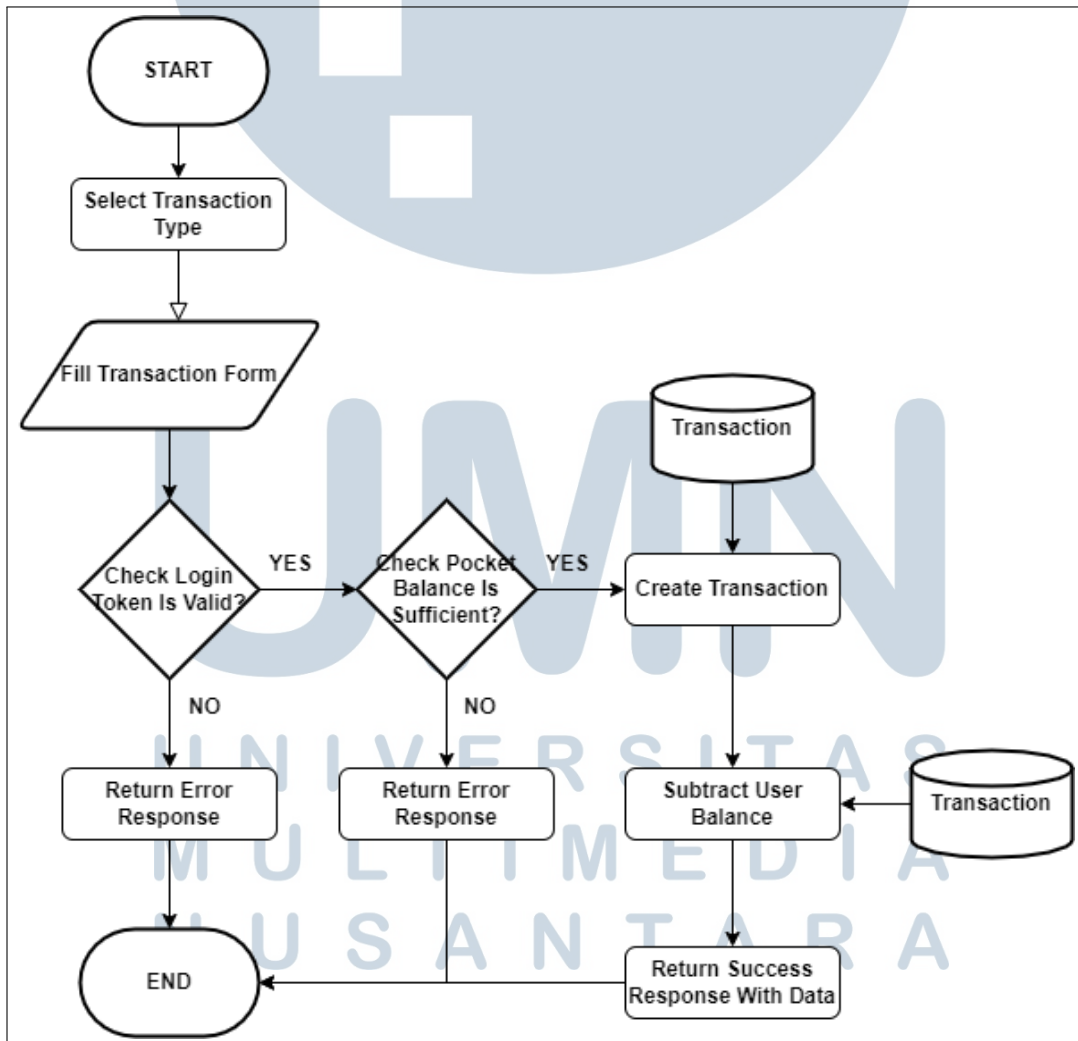
Untuk melakukan pemberian akses untuk subpocket yang telah dibuat kepada user lain di perlukan parameter seperti userid, dan pocketid.

```
... "options": {
... |... "transaction_type": "REQUEST"
... }
}
```

Gambar 3.18. Rancangan Parameter Transaction Index

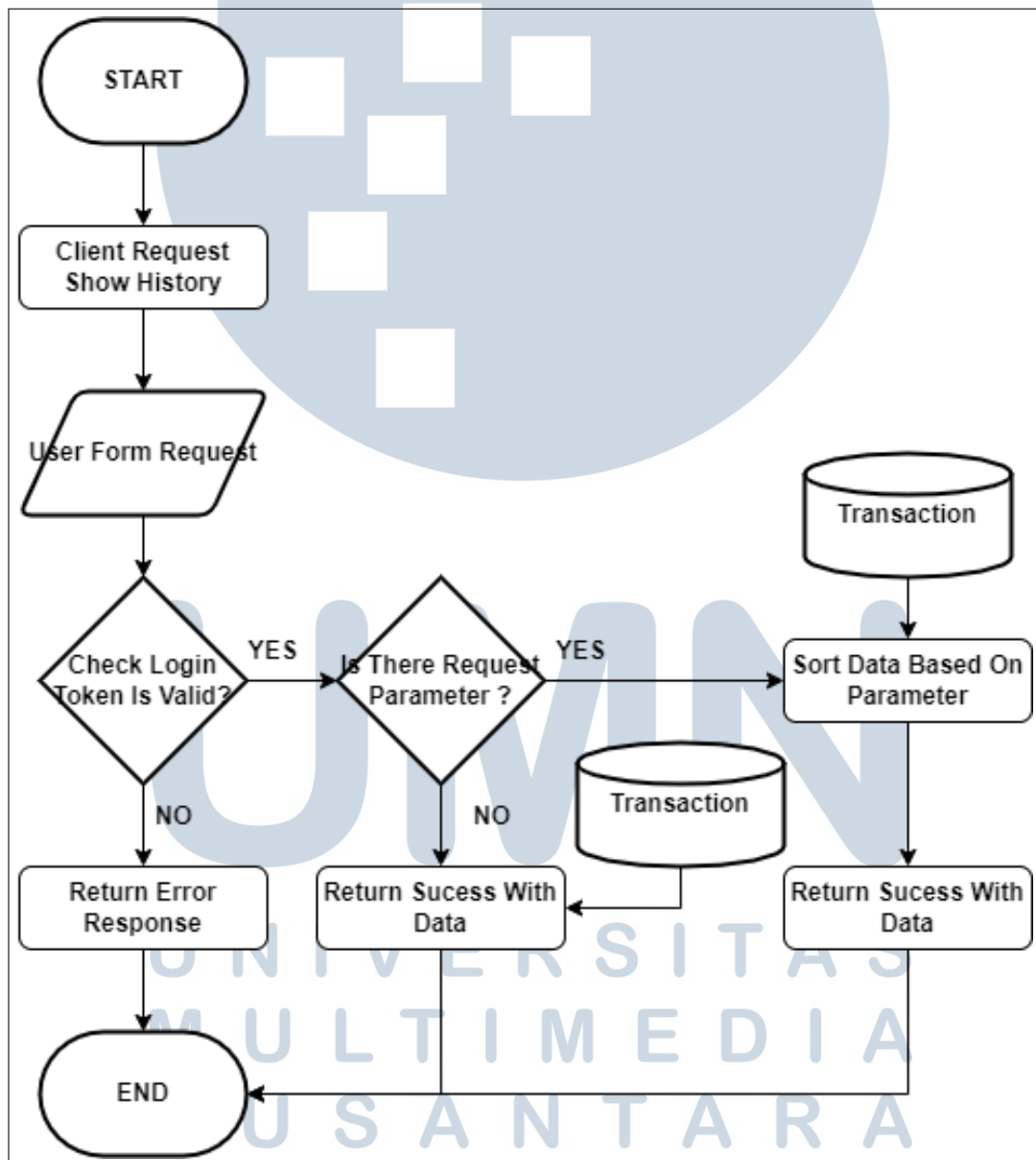
Parameter yang dapat dimasukkan pada field transactiontype adalah REQUEST, PAY, MOVE, SEND. ini akan mengeluarkan data sesuai dengan transactiontype yang diinginkan oleh pengguna.

#### D. Perancangan Flowchart



Gambar 3.19. Rancangan Flowchart Transaction

Alur yang terjadi pada saat dilakukannya suatu transaksi adalah ketika pengguna melakukan transaksi pada sistem akan terjadi pemrosesan berupa pengecekan token yang dimiliki oleh pengguna valid, form sudah terisi semua, dan dana pengguna mencukupi. Setelah pengecekan selesai dilakukan data transaksi akan disimpan kedalam *database transaction* lalu melakukan pengurangan dana yang dimiliki pengguna pada *database pocket*.



Gambar 3.20. Rancangan Flowchart History

Ketika pengguna ingin melihat daftar transaksi apa saja yang



pernah dilakukan pengguna akan mengirim *form* kepada *server* untuk selanjutnya dilakukan pengecekan token yang dimiliki oleh pengguna valid, dan apakah ada parameter pada permintaan yang dikirimkan. jika sudah di validasi *server* akan mengembalikan data sesuai dengan permintaan yang diinginkan pengguna.

### 3.3.3 Implementasi

```

"success": {
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiJ1IiwianRpIjoiaWZ3NzVnZTNkYTExOjE2MDYyZmYyZmZkNGM1MTI0ThhMzhjMGU4NWRIYmQyOTYyZjYyY2E0TEIzGVjMDE5MDc0ODY3Nz1iMTMlLCJpYXZlbnVzIjE2MDYyZmYyZmZkNGM1MTI0ThhMzhjMGU4NWRIYmQyOTYyZjYyY2E0TEIzGVjMDE5MDc0ODY3Nz1pbXQ9LjJn324awq6Va0iwC4nUipwIgdMjHcg93pHWd8-xemIP4NIJhvVxwFXl0saH8G2HHnSCpev5455-ftvFU8b0X-uTJWxwd8-Xo7SCsqAqum-wdNsyjedGNw-g-MK75mLZmwn1TBUbzdb3858w3_I6I7uFchFLV28L2czBpuq7TBy0FgtcI2GMe9xMzjcIo0iHGFiBXwk-16ngxvg3hpmRoedjnohU7PZz_7Dj4BTB6m7a6GB0hEY8iRYn_1419sV3sIPaVxpIDy8DV0YF8NXw0NqKu8wdkXy2ixJfjM0q_PkEwdtYqRLZ3nnI5BelhNR1zguJV0M098Ij5PwHMua3S-8TnXL-JkSE3y6iPck2uXoSgZaqI21QZzBJxKlPriz0FFPaujy_uqOxYbt2300oJZDwMKYKf1UHDI3hcS-uQv_9HW2XZdG82m_w8IcfnhFGDfpIonV3Rnn9TSBozwwAmoudSMQ0e96nLySG0xIXLpezT8RjeMcIv9s1tMEUMtXZgec9m-JZCfveAH1kVvC13P-HFeORjQA4MFTzQevq90RLp-9zEe0CZHUTprYXw1u6spqK9UCL1pADjJsrGLd-URLku627jbyP8HXK0izptfP7EmLycHn09PWOx0UsNv5dFrFibvG12P161m1pM-RBNKYE13ghU08vI"
}
    
```

Gambar 3.21. Implementasi API Auth Login

Gambar 3.21 merupakan implementasi yang dilakukan ketika login, setelah pengguna berhasil melakukan login. *server* akan mengembalikan sebuah *Bearer Token* yang dapat digunakan oleh pengguna sebagai identifikasi/otentikasi untuk mengakses fitur - fitur yang tersedia pada aplikasi *Digital Pocket API* yang telah dibangun. Karena jika tidak memiliki *Bearer Token* pengguna tidak dapat mengakses fitur - fitur yang tersedia pada aplikasi *Digital Pocket API*, dikarenakan adanya otentikasi pengguna untuk setiap fitur yang tersedia pada aplikasi *Digital pocket API*.

```

{
  "message": "Password mismatch"
}
    
```

Gambar 3.22. Implementasi API Auth Wrong Password

Gambar 3.22 adalah implementasi pencocokan data pengguna yang digunakan untuk login, pada saat data pengguna yang dikirimkan tidak ditemukan pada *server* atau salah memasukkan kata sandi *server* akan memberikan *respond* dalam bentuk gambar 3.22. Digunakan untuk memberi tahu pengguna bahwa data yang diberikan oleh pengguna salah.

```
{
  "status": {
    "code": 401,
    "message": "Unauthenticated."
  },
  "data": null
}
```

Gambar 3.23. Implementasi API Auth

Gambar 3.23 adalah fitur pengecekan otentikasi yang digunakan pada setiap fitur yang tersedia pada aplikasi *Digital Pocket API* yang akan melakukan pengecekan pada *Bearer Token* yang digunakan oleh pengguna untuk mengakses fitur yang tersedia. Jika *Bearer Token* yang diberikan oleh pengguna salah ataupun tidak diberikan *server* akan memberikan *respond* seperti pada gambar 3.23 untuk memberi tahu pengguna bahwa pengguna tidak memiliki izin untuk mengakses fitur tersebut.

```
"Message": "Please verified your account for more action"
```

Gambar 3.24. Implementasi API Auth Email Unverified

Gambar 3.24 merupakan implementasi untuk melakukan pengecekan verifikasi alamat surel pengguna apakah sudah di verifikasi atau belum fitur ini mengacu pada kebutuhan yang diberikan perusahaan agar setiap pengguna yang menggunakan aplikasi *Digital Pocket API* melakukan verifikasi alamat surel terlebih dahulu untuk dapat menggunakan fitur - fitur yang tersedia. Ketika ditemukan bahwa pengguna belum melakukan verifikasi alamat surel, *respond* seperti pada gambar 3.24 lah yang akan diberikan oleh *server* untuk memberi tahu pengguna.

```

{
  "status": {
    "code": 200,
    "message": "Success"
  },
  "data": [
    {
      "transaction_id": 3,
      "user_id": 2,
      "status_id": null,
      "source_type": null,
      "destination_type": "pockets",
      "transaction_type": "TOPUP",
      "transaction_desc": "Top Up Balance",
      "transaction_date": "2021-10-25 10:42:16",
      "nominal": "20000",
      "total": "20000"
    },
    {
      "transaction_id": 4,
      "user_id": 2,
      "status_id": null,
      "source_type": null,

```

Gambar 3.25. Implementasi Transaction Index

Gambar 3.25 adalah *respond* yang diberikan *server* ketika pengguna mengakses fitur *Transaction Index* yang merupakan fitur untuk melakukan pengecekan daftar transaksi yang telah dilakukan pengguna dalam aplikasi *Digital Pocket API*. Fitur ini membantu pengguna dalam melakukan pengecekan transaksi apa saja yang telah dilakukan pengguna dalam menggunakan aplikasi *Digital Pocket API* dan pengguna juga dapat mengetahui jika sewaktu - waktu terjadi keanehan pada dompet milik pengguna.



```
    "status": {
      "code": 201,
      "message": "Creating data successful"
    },
    "data": {
      "transaction_id": 61,
      "user_id": 2,
      "status_id": 1,
      "source_type": null,
      "destination_type": "pockets",
      "transaction_type": "TOPUP",
      "transaction_desc": "Top Up Balance",
      "transaction_date": {
        "date": "2021-12-16 12:15:49.942254",
        "timezone_type": 3,
        "timezone": "Asia/Jakarta"
      },
      "nominal": "10000000",
      "total": "10000000"
    }
  }
}
```

Gambar 3.26. Implementasi Transaction Topup

Gambar 3.26 merupakan implementasi fitur *Transaction Topup* yang terdapat pada aplikasi *Digital Pocket API* yang berguna untuk menambahkan dana pada dompet yang dimiliki oleh pengguna, pada fitur *Transaction Topup* terdapat keterangan berupa tanggal pengisian, nominal pengisian dana, dan tujuan pengisian dana.

Kemudian transaksi tersebut akan dicatatkan kedalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

```
"status": {
  "code": 201,
  "message": "Creating data successful"
},
"data": {
  "transaction_id": 62,
  "user_id": 2,
  "status_id": 1,
  "source_type": "pockets",
  "destination_type": "pockets",
  "transaction_type": "SEND",
  "transaction_desc": "Jajan Bulanan, Bulan October",
  "transaction_date": {
    "date": "2021-12-16 12:24:44.864984",
    "timezone_type": 3,
    "timezone": "Asia/Jakarta"
  },
  "nominal": "20000",
  "total": "20000"
}
```

Gambar 3.27. Implementasi Transaction Send

Gambar 3.27 fitur *Transaction Send* yang berfungsi untuk mengirimkan dana yang terdapat dalam dompet pribadi pengguna kepada pengguna lain yang menggunakan aplikasi *Digital Pocket API*, pada fitur *Transaction Send* terdapat keterangan berupa status transaksi, sumber dana, dompet tujuan, tanggal transaksi, keterangan transaksi, dan nominal transaksi.

Kemudian transaksi tersebut akan dicatatkan kedalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```
{
  "status": {
    "code": 201,
    "message": "Creating data successful"
  },
  "data": {
    "transaction_id": 63,
    "user_id": 2,
    "status_id": 1,
    "source_type": "pockets",
    "destination_type": "pockets",
    "transaction_type": "REQUEST",
    "transaction_desc": "Minta Uang Jajan Mak",
    "transaction_date": {
      "date": "2021-12-16 12:28:36.268547",
      "timezone_type": 3,
      "timezone": "Asia/Jakarta"
    },
    "nominal": "20000",
    "total": "20000"
  }
}
```

Gambar 3.28. Implementasi Transaction Request

Gambar 3.28 fitur *Transaction Request* ini berfungsi untuk melakukan pengajuan permintaan dana kepada pengguna lain yang menggunakan aplikasi *Digital Pocket API* tetapi dana tidak akan langsung diterima oleh pengguna yang melakukan permintaan karena harus adanya persetujuan dari pihak yang dituju oleh permintaan tersebut. Pada fitur *Transaction Request* terdapat keterangan berupa status transaksi, sumber dana, dompet tujuan, tanggal transaksi, keterangan transaksi, dan nominal transaksi.

Kemudian transaksi tersebut akan dicatatkan kedalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

```
"status": {
  "code": 201,
  "message": "Creating data successful"
},
"data": {
  "transaction_id": 64,
  "user_id": 2,
  "status_id": 1,
  "source_type": "pockets",
  "destination_type": "merchant",
  "transaction_type": "PAY",
  "transaction_desc": "Bayar Bakso Di Kang Bankso",
  "transaction_date": {
    "date": "2021-12-16 12:31:00.344176",
    "timezone_type": 3,
    "timezone": "Asia/Jakarta"
  },
  "nominal": "20000",
  "total": "20000"
}
```

Gambar 3.29. Implementasi Transaction Pay

Gambar 3.29 adalah fitur *Transaction Pay* fitur tersebut berfungsi untuk melakukan pembayaran kepada *merchant*, *vendor*, maupun *multifinance*, yang telah melakukan kerjasama dengan perusahaan untuk mempermudah melakukan transaksi dengan pihak - pihak tersebut. Pada fitur *Transaction Pay* terdapat keterangan berupa status transaksi, sumber dana, tujuan transaksi, tanggal transaksi, keterangan transaksi, dan nominal transaksi.

Kemudian transaksi tersebut akan dicatatkan kedalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



```
"status": {
  "code": 201,
  "message": "Creating data successful"
},
"data": {
  "transaction_id": 65,
  "user_id": 2,
  "status_id": 1,
  "source_type": "pockets",
  "destination_type": "subpockets",
  "transaction_type": "MOVE",
  "transaction_desc": "Minta Uang Jajan Mak",
  "transaction_date": {
    "date": "2021-12-16 12:35:44.175416",
    "timezone_type": 3,
    "timezone": "Asia/Jakarta"
  },
  "nominal": "20000",
  "total": "20000"
}
```

Gambar 3.30. Implementasi Transaction Move

Gambar 3.30 merupakan fitur *Transaction Move* ini berfungsi untuk melakukan pemindahan dana dari dompet utama ke dalam dompet cadangan yang telah dibuat oleh pengguna maupun dompet cadangan yang dapat diakses oleh pengguna, pada fitur *Transaction Move* terdapat keterangan berupa status transaksi, sumber dana, dompet tujuan, tanggal transaksi, keterangan transaksi, dan nominal transaksi.

Kemudian transaksi tersebut akan dicatatkan ke dalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```
"status": {
  "code": 201,
  "message": "Creating data successful"
},
"data": {
  "transaction_id": 67,
  "user_id": 2,
  "status_id": 1,
  "source_type": "pockets",
  "destination_type": "pockets",
  "transaction_type": "SEND",
  "transaction_desc": "Maap Cuma Bisa Kasih Segini",
  "transaction_date": {
    "date": "2021-12-16 12:40:00.381959",
    "timezone_type": 3,
    "timezone": "Asia/Jakarta"
  },
  "nominal": "50000",
  "total": "50000"
}
```

Gambar 3.31. Implementasi Transaction Accept Request

Gambar 3.31 fitur *Transaction Accept Request* berfungsi untuk menerima permintaan dana dari pengguna lain yang telah melakukan permintaan melalui fitur *Transaction Request* kepada pengguna, lalu pengguna akan memutuskan apakah akan menerima permintaan ataupun menolak permintaan tersebut. Pada fitur *Transaction Accept Request* ini terdapat keterangan berupa status transaksi, sumber dana, dompet tujuan, tanggal transaksi, keterangan transaksi, dan nominal transaksi.

Kemudian transaksi tersebut akan dicatatkan kedalam daftar transaksi yang telah dilakukan oleh pengguna untuk melakukan pencatatan yang sewaktu - waktu dapat dilihat.

```
"status": {
  "code": 201,
  "message": "Creating data successful"
},
"sub_pocket": {
  "id": 2,
  "balance": 0,
  "type": "Shared",
  "wallet_name": "Tabungan Bersama",
  "shared_user": [
    {
      "name": "asda",
      "email": "asda@gmail.com"
    }
  ]
},
"user_pocket": {
  "user_id": 2,
  "sub_pocket_id": 2
}
```

Gambar 3.32. Implementasi Sub Pocket Create

Gambar 3.32 merupakan fitur *Sub Pocket Create* berfungsi untuk membuat dompet cadangan yang diinginkan pengguna, dompet cadangan tersebut akan langsung diatur oleh pengguna apakah dompet cadangan tersebut dapat dibagikan ataupun sebagai penyimpanan pribadinya saja.

```
"Message": "User Have Been Permitted To Access The Sub Pocket!"
```

Gambar 3.33. Implementasi Sub Pocket Link

Gambar 3.33 fitur *Sub Pocket Link*, fitur tersebut berfungsi untuk memberikan akses dompet cadangan yang dimiliki pengguna kepada pengguna lain. Fitur tersebut dibuat berdasarkan kebutuhan perusahaan, karena dalam perancangan aplikasi *Digital Pocket API* merasakan perlu adanya dompet yang dapat dibagikan dan tidak mengganggu dompet pribadi masing-masing pengguna.

```
"status": {
  "code": 200,
  "message": "Success"
},
"main_pocket": {
  "id": 2,
  "unique_id": "8c5162438d8baea83549824ddedd44d2",
  "user_id": 2,
  "balance": "21020000"
}
```

Gambar 3.34. Implementasi Main Pocket Index

Gambar 3.34 adalah fitur *Main Pocket Index* fitur ini berfungsi untuk melakukan pengecekan pada dompet utama user, dalam fitur tersebut terdapat keterangan berupa *Unique Id* yang akan menjadi tanda pengenal dalam melakukan setiap transaksi, bukan hanya *Unuqie Id* saja. Pada fitur *Main Pocket Index* juga menampilkan jumlah dana yang dimiliki pengguna.

```
"status": {
  "code": 200,
  "message": "Success"
},
"sub_pocket": [
  {
    "id": 1,
    "balance": "20000.00",
    "type": "Shared",
    "wallet_name": "Tabungan Bersama",
    "shared_user": [
      {
        "name": "asd",
        "email": "asd@gmail.com"
      },
      {
        "name": "asda",
        "email": "asda@gmail.com"
      }
    ]
  }
]
```

Gambar 3.35. Implementasi Sub Pocket Index

Gambar 3.35 fitur *Sub Pocket Index* berfungsi untuk mengecek dom-

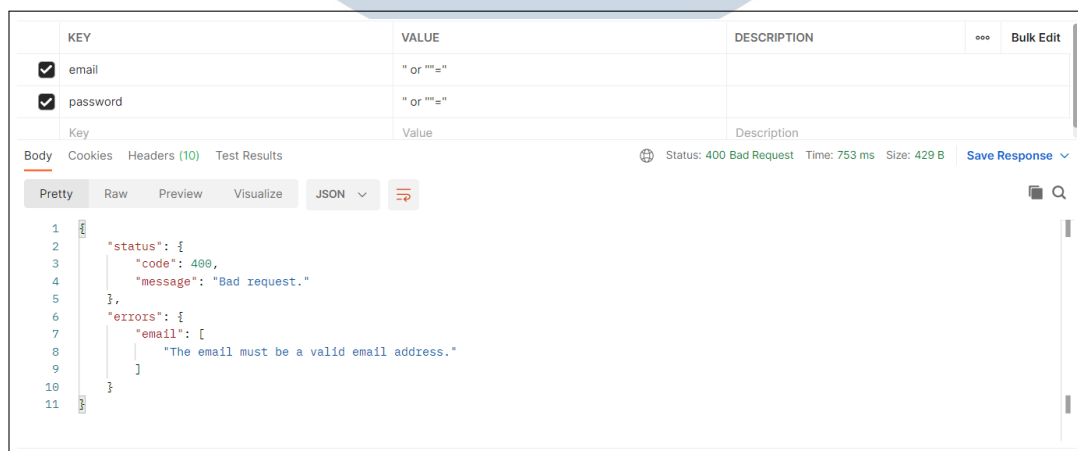
pet cadangan pengguna yang akan menampilkan dana dan pengguna mana saja yang mendapatkan diizinkan untuk mengakses dompet cadangan tersebut.

### 3.4 Pengujian

Tahapan pengujian dilakukan untuk mengetahui apakah aplikasi sudah berjalan sesuai dengan apa yang diprogramkan dan tidak adanya celah pada program, berikut adalah pengujian yang dilakukan pada aplikasi *Digital Pocket API*.

#### 3.4.1 SQL Injection

SQL Injection merupakan teknik eksploitasi dengan cara memodifikasi perintah sql pada form input aplikasi yang memungkinkan penyerang untuk dapat mengirimkan sintaks ke database aplikasi. SQL Injection juga dapat didefinisikan sebagai teknik eksploitasi celah keamanan pada layer database untuk mendapatkan query data pada sebuah aplikasi.



Gambar 3.36. Pengujian SQL Injection

Gambar 3.36 adalah hasil pengujian untuk *SQL Injection* pada aplikasi *SQL Injection* sudah terdapat validasi untuk alamat surel yang dimasukkan haruslah berupa alamat surel yang resmi. Bukan hanya pada fitur login saja melainkan pada setiap fitur yang terdapat pada aplikasi *Digital Pocket API* sudah terdapat validasi input yang dapat meminimalisir terjadinya *SQL Injection*.

## **3.5 Kendala dan Solusi yang Ditemukan**

### **3.5.1 Kendala yang Ditemukan**

Berikut adalah beberapa kendala yang dialami selama proses pengerjaan magang.

- Pembangunan API menggunakan PHP dengan framework laravel.
- Penggunaan software manajemen proyek GitLab dan Fork yang belum pernah digunakan sebelumnya.
- Kekurangpahaman akan proses pembangunan API dengan menggunakan standarisasi kode yang ada pada perusahaan.

### **3.5.2 Solusi atas Kendala yang Ditemukan**

Berikut adalah beberapa solusi atas kendala yang dialami

- Mempelajari framework melalui video tutorial di youtube, membaca dokumentasi, dan meminta referensi dari senior dan rekan kerja.
- Mencoba untuk membiasakan diri dengan GitLab dan membuat proyek / repository untuk melakukan uji coba Fork.
- Menanyakan kepada senior dari rekan kerja tentang fungsi yang tidak di mengerti.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A