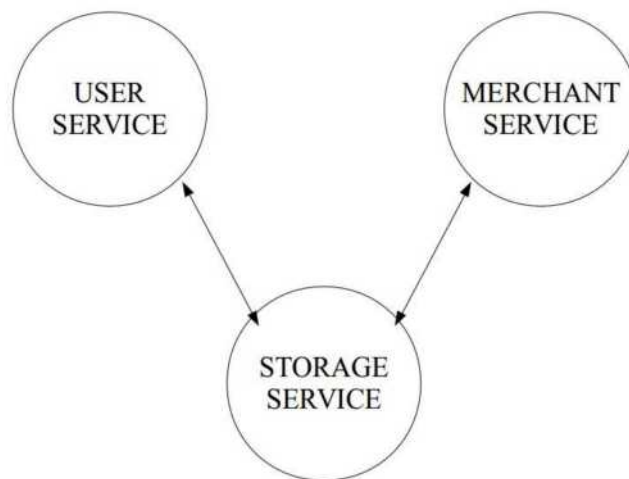


BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Desain Web Service

Pada sistem Tukutu terdapat 3 service yang digunakan oleh platform Android, tiap *service* memiliki fungsionalitas tersendiri dalam menangani *resource*. Berikut service yang digunakan:



Gambar 4.1: Desain Web Service

4.1.1 User Service

Service ini memiliki fungsionalitas yang digunakan oleh pembeli, karena memuat fitur – fitur yang diperlukan dalam transaksi pada aplikasi Tukutu. Service ini bertugas memberikan informasi yang ada pada aplikasi Tukutu seperti daftar sepatu yang dijual, brand sepatu, akun pengguna, dan transaksi. *User service* dibangun menggunakan framework Lumen. Berikut fitur yang ada:

- Order (pembelian)
- Bid (penawaran)

- Diskusi
- Manajemen Transaksi

4.1.2 Merchant Service

Merchant service memiliki fungsionalitas untuk memanajemen penjualan sepatu pada aplikasi Tukutu, *service* ini hanya bisa digunakan oleh akun yang telah terdaftar sebagai penjual kecuali pada fitur pengajuan sebagai *merchant* (penjual) pada aplikasi. *Merchant service* dibangun menggunakan framework Lumen. Berikut fitur yang ada:

- Gabung jadi penjual
- Manajemen Sepatu
- Manajemen Transaksi

4.1.3 Storage Service

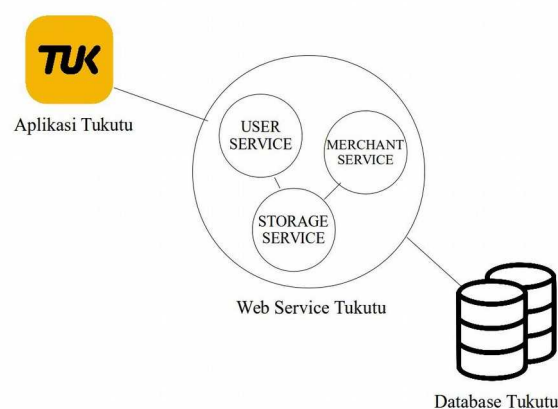
Storage service berguna sebagai tempat menyimpan gambar dan menjalankan fungsi tambahan yang bersangkutan dengan plugin pihak ketiga atau bawaan milik framework Laravel. *Storage service* terhubung dengan *user service* dan *merchant service*, ketika resource yang di akses pada *user service* atau *merchant service* mengandung fungsi pihak ketiga yang harus dijalankan maka akan langsung menghubungi *storage service* untuk menjalankannya. Hal ini dilakukan karena framework Laravel lebih mudah dalam pemasangan plugin pihak ketiga. Berikut fungsionalitas yang terdapat pada *storage service*:

- Menyimpan gambar
- Mengirim email SMTP kepada pembeli dan penjual
- Mengirim notifikasi FCM (Firebase Cloud Messaging) kepada pembeli dan penjual

- Menjalankan sistem *auto cancel* (pembatalan otomatis) berdasarkan waktu yang telah ditetapkan yaitu
 - Waktu pembayaran untuk pembelian : 3 Jam
 - Pengisian form pembelian jika penawarannya diterima : 12 Jam
 - Waktu kirim merchant (penjual) ke warehouse Tukutu : 2 Hari
 - Waktu merchant (penjual) mencairkan dana : 1 Hari

4.2 Alur Kerja Web Services

Pada aplikasi Tukutu *service* yang diakses pertama adalah *user service* karena pada *service* ini memuat fitur utama pada Tukutu. Kemudian pada *merchant service* digunakan pada saat ada akun yang ingin mendaftar menjadi penjual dan manajemen penjualan pada aplikasi Tukutu. Kedua *service* tersebut terhubung pada satu *service* yaitu *storage service* yang merupakan tempat untuk menyimpan *resource* dan *plugin* pihak ketiga. Data *resource* yang disimpan tidak hanya gambar, tapi juga *database migrations* yang merupakan bawaan milik Laravel. *Database migrations* kemudian di *generate* menjadi tabel pada database yang kemudian diakses *user service*, *merchant service*, dan *storage service*.



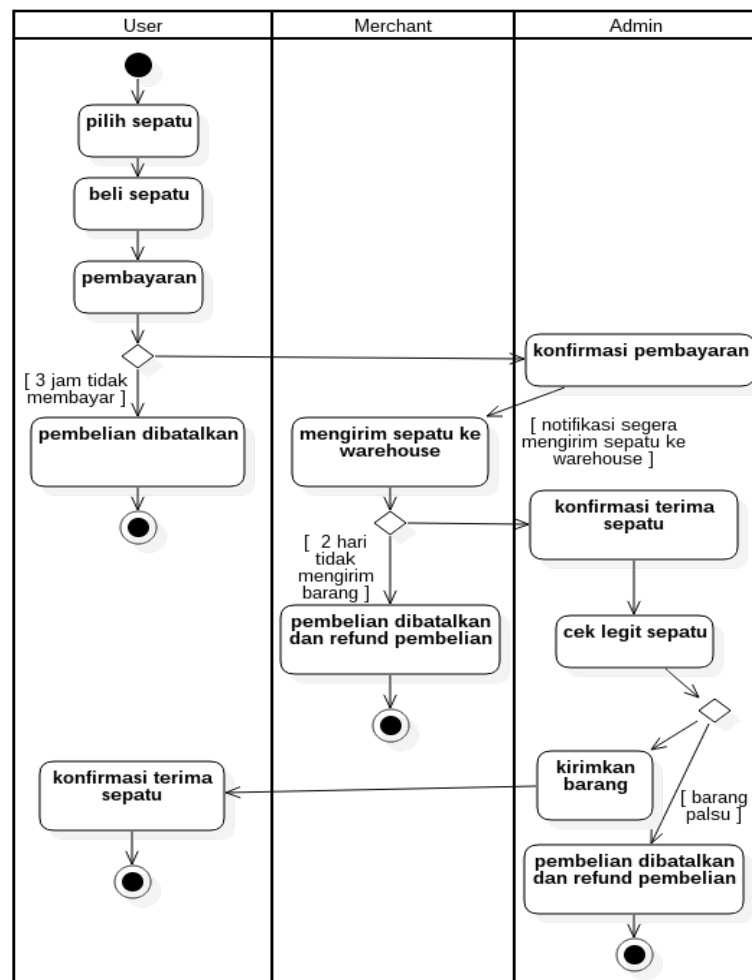
Gambar 4.2: Alur Web Service Tukutu

4.3 Proses Transaksi

Pada Tukutu terdapat 2 cara dalam transaksi, yaitu pembelian dan penawaran. Berikut penjelasannya:

4.3.1 Pembelian

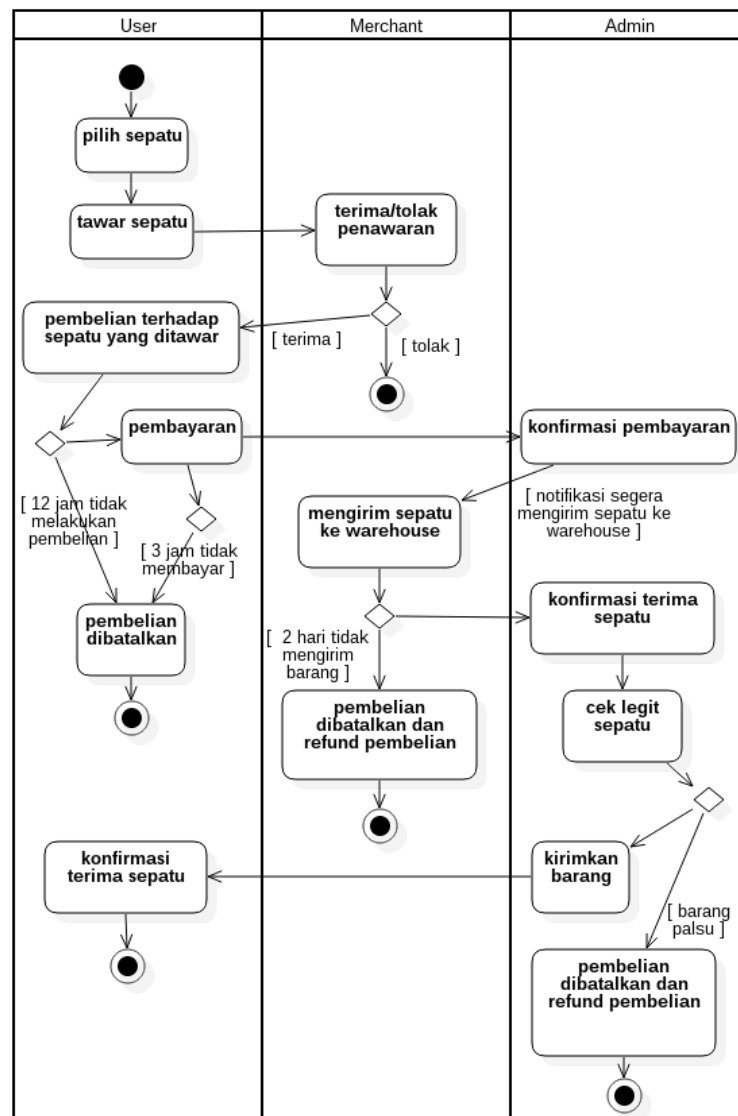
Pembelian dilakukan pada sepatu yang telah dipilih, kemudian pembeli mengisi data pembelian. Pembelian dapat dibatalkan jika pembeli tidak segera membayar dalam waktu 3 jam, penjual tidak mengirim barang ke warehouse dalam waktu 2 hari, dan barang terbukti palsu.



Gambar 4.3: Alur Pembelian

4.3.2 Penawaran

Pada penawaran pembeli dapat menawar sepatu yang dipilih jika sepatu dapat ditawarkan, jika penawaran diterima maka pembeli harus melakukan pembelian sebelum 12 jam jika tidak ingin kehilangan hasil tawarnya. Alur pembelian setelah tawar sama seperti alur pembelian sebelumnya.



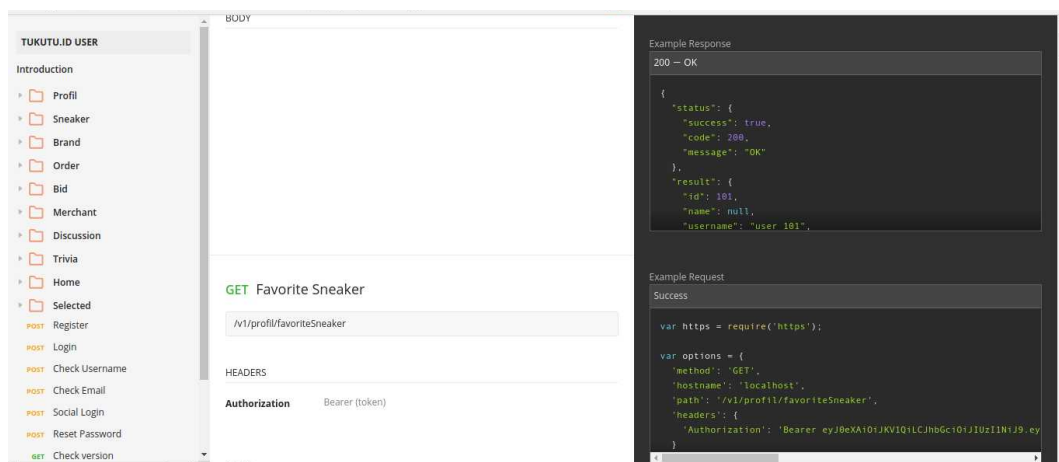
Gambar 4.4: Alur Penawaran

4.4 Implementasi dan Pembahasan

Berikut hasil dari implentasi *web service* pada Tukutu.

4.4.1 Dokumentasi API

Untuk memudahkan integrasi *web service* ke aplikasi Tukutu, dibutuhkan dokumentasi API yang memuat daftar *endpoint* API dan data *response* yang diterima.



Gambar 4.5: Dokumentasi API

Dari dokumentasi tersebut bagian developer platform Android Tukutu dapat mengetahui data yang diterima. Data yang diterima dalam bentuk JSON sesuai dengan standar format teks yang telah ditentukan sebelumnya pada **sub bab 3.5** di atas. Jika hasil *request* sukses data *resource* terdapat pada objek *result* yang berisi data *object* atau *array* yang berisikan informasi untuk ditampilkan pada sisi *client*.

4.4.2 Response

Standar format JSON yang digunakan yang sesuai dengan gambar (gambar 3.6) yang berisikan informasi objek status dan result. Objek status berisi informasi dari hasil request yang terdiri dari success, code, dan message yang

telah dideskripsikan pada sub bab 3.5, sedangkan objek result berisikan data resource yang dibutuhkan oleh aplikasi Android.

```
{
  "status": {
    "success": true,
    "code": 200,
    "message": "message response"
  },
  "result": {
    "object_1": "value object 1",
    "object_2": "value object 2",
    "object_3": "value object 3",
    "object_4": "value object 4"
  }
}
```

Gambar 4.6: Contoh response

```
{
  "status": {
    "success": true,
    "code": 200,
    "message": "message response"
  },
  "result": {
    "object_1": "value object 1",
    "object_2": "value object 2",
    "object_3": "value object 3",
    "object_4": "value object 4",
    "objcet_5": {
      "object_6": "value object 6"
    }
  }
}
```

Gambar 4.7: Contoh response objek didalam objek

```
{
  "status": {
    "success": true,
    "code": 200,
    "message": "message response"
  },
  "result": [
    {
      "object_1": "value object 1"
    },
    {
      "object_2": "value object 2"
    }
  ]
}
```

Gambar 4.8: Contoh response dengan data array

4.4.3 Autentikasi

Proses autentikasi Tukutu menggunakan JSON Web Token (JWT), dimana pembuatan token terjadi saat aplikasi Tukutu (*client*) mengakses fungsi login pada web service. Hasil generate token kemudian disimpan pada sisi *client*, token tersebut digunakan pada setiap kali request/mengakses web service untuk meminta, mengirim atau memanipulasi data resource.

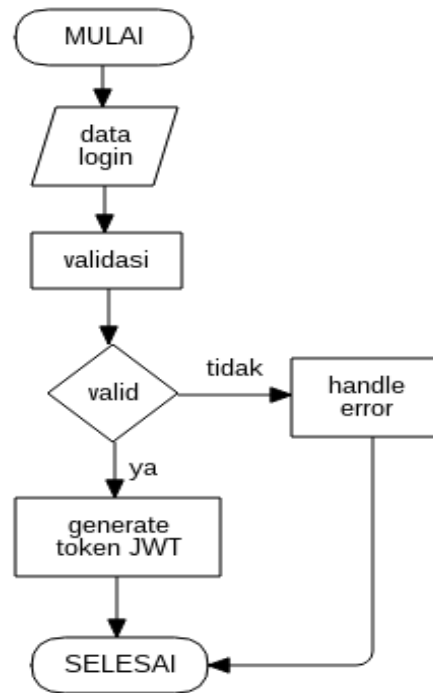
```
"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ1dWtldHUuSUQiLCJpYXQiOiJlMzI5NDQ3Njg2Nywic3ViIjozMjUzMDUzImp0aSI6IksZsT2ImckN0ODIifQ.fbg20mnSFPncbGC2qdvPH51XaSPou0_OCt4K18svrQ"
```

Gambar 4.9: Contoh token JWT

Hasil token yang dibuat kemudian disimpan juga pada database untuk pengecekan jika hasil token tersebut milik Tukutu karena digunakan sebagai hak akses ke data resource. Token JWT yang didapat hanya bisa digunakan pada *user service* dan *merchant service*.

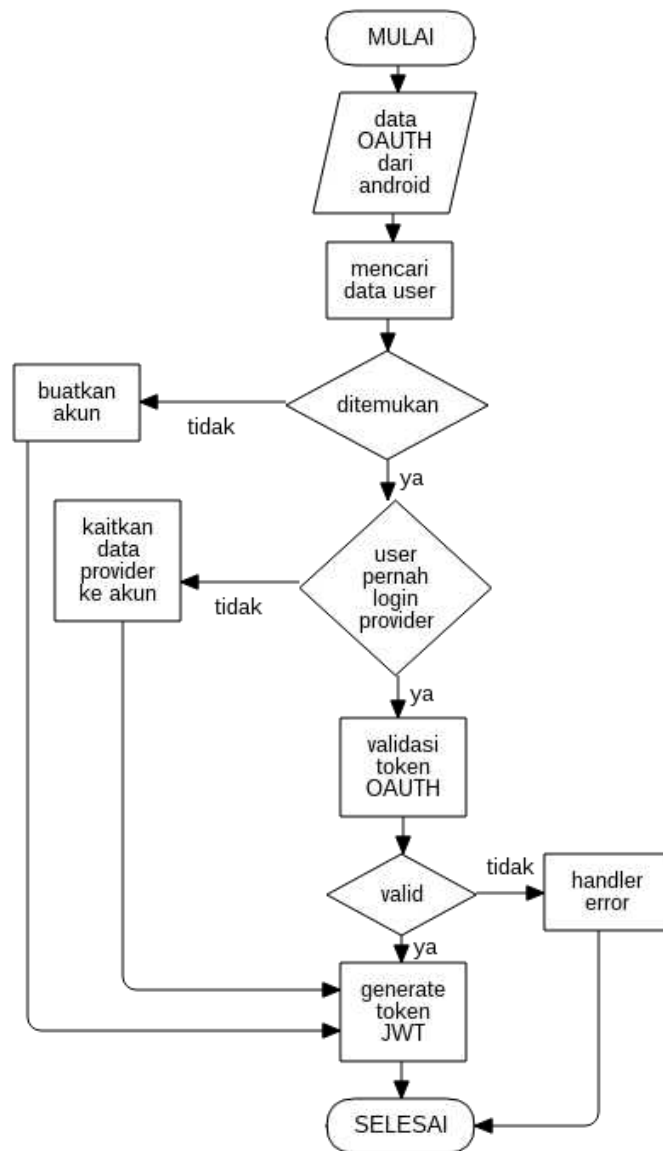
4.4.4 Flowchart

a) Login



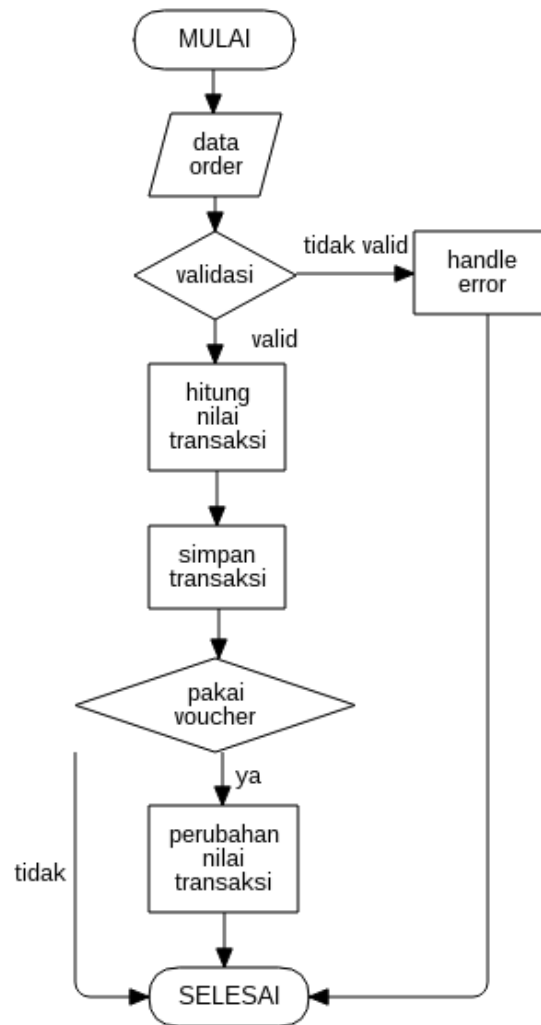
Gambar 4.10: Flowchart Login

b) Login Provider



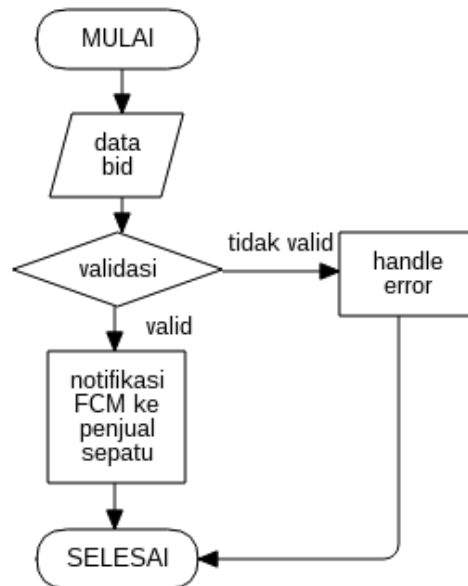
Gambar 4.11: Flowchart Login Provider

c) Order (pembelian)



Gambar 4.12: Flowchart Order

d) Bid (penawaran)



Gambar 4.13: Flowchart Bid

Jika nilai tawar diterima oleh penjual maka pembeli dapat melakukan pembelian dengan alur pembelian yang sama pada *flowchart* sebelumnya yaitu *flowchart order*.

e) Diskusi



Gambar 4.14: Flowchart Diskusi

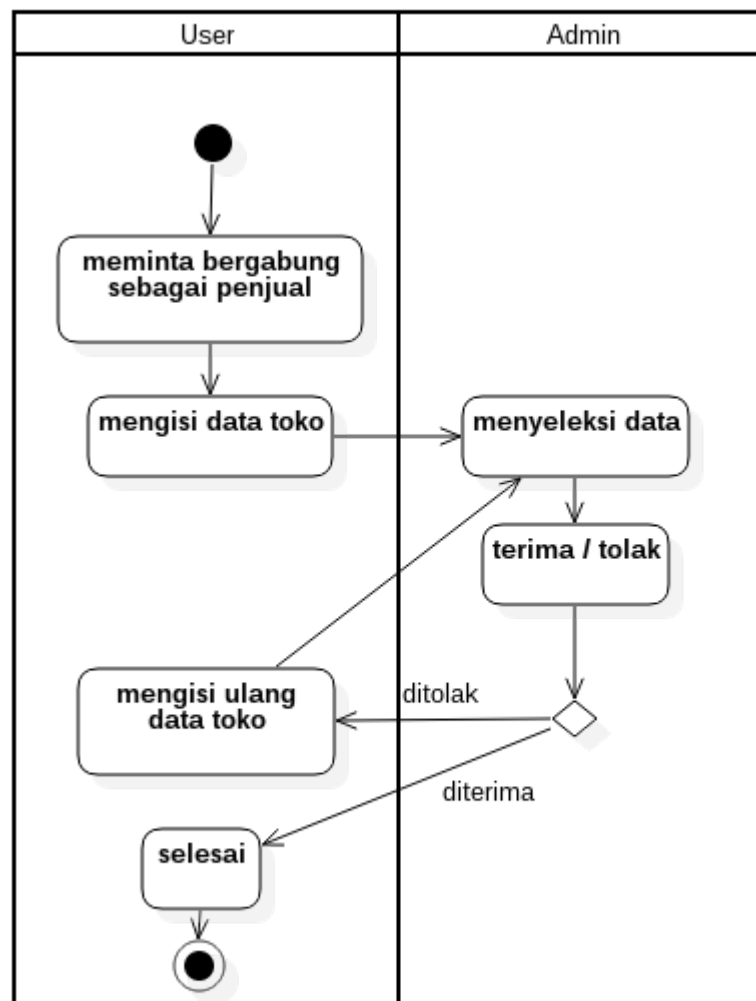
f) Balas Diskusi



Gambar 4.15: Flowchart Balas Diskusi

4.4.5 Merchant

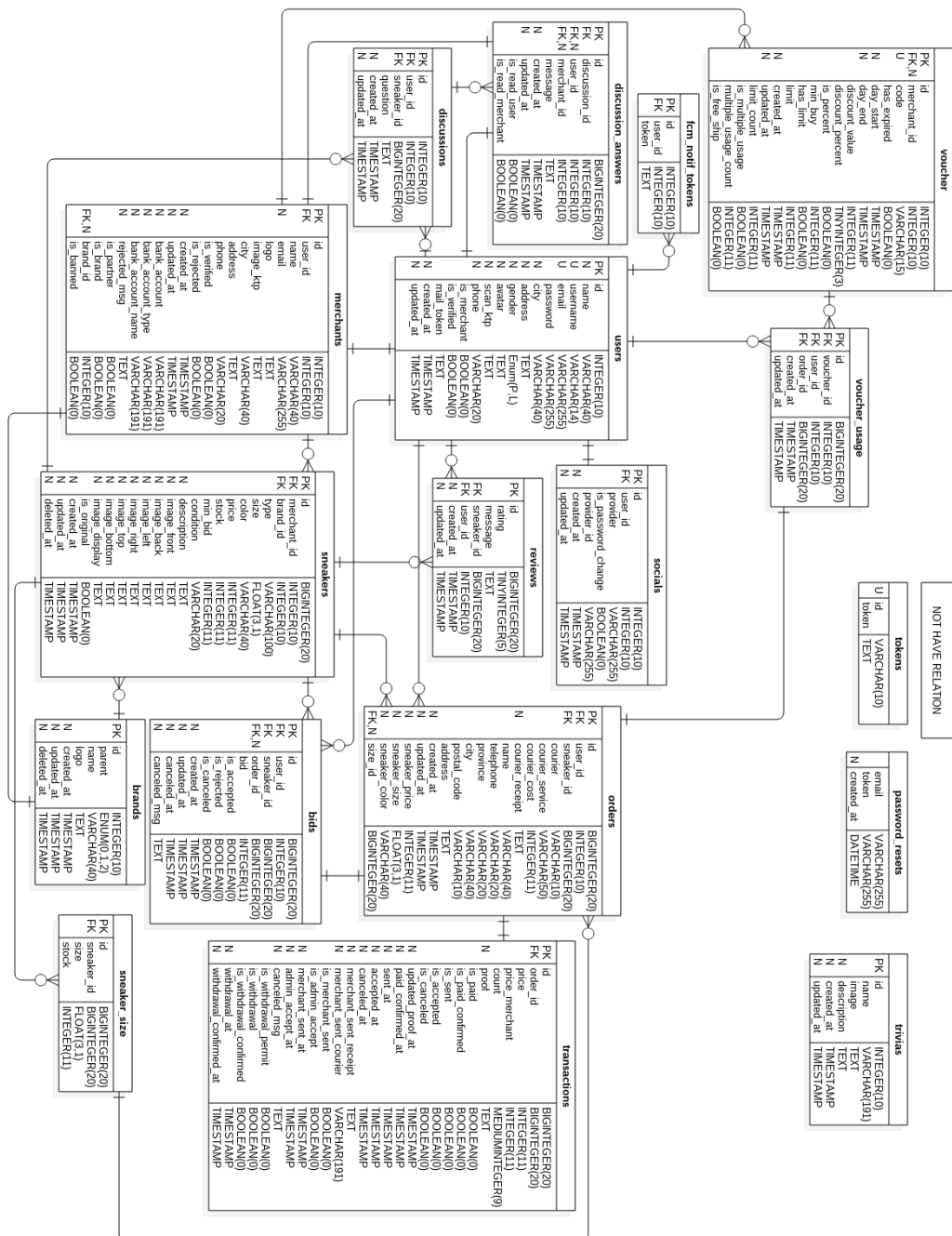
Penjual pada Tukutu disebut merchant, setiap akun dapat menjadi penjual dengan syarat dan ketentuan yang berlaku. Untuk menjadi penjual akun yang terdaftar harus sudah terferifikasi, kemudian mengajukan data – data yang diperlukan untuk diseleksi oleh Admin Tukutu.



Gambar 4.16: Alur jadi Merchant

Setelah menjadi Merchant user dapat mulai menambah sepatunya di Tukutu dan menjualnya.

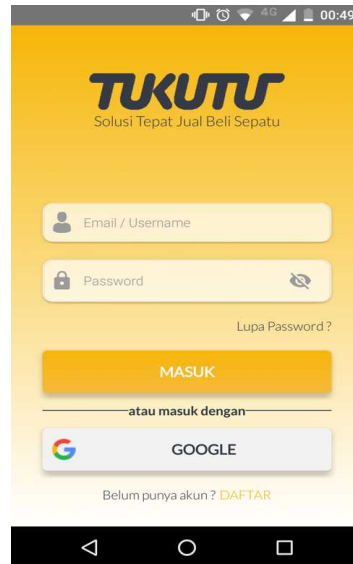
4.5 Perancangan Database



Gambar 4.17: ERD Tukutu

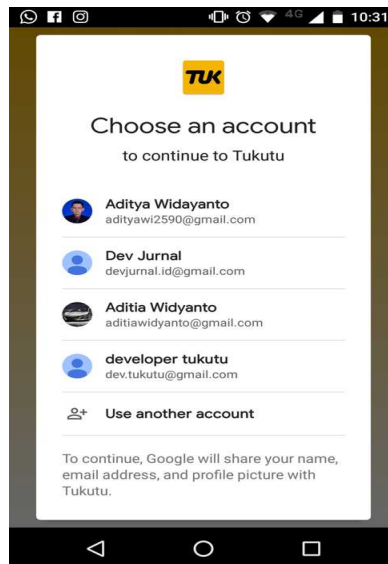
4.6 Implementasi Aplikasi Tukutu (Client)

4.6.1 Halaman Login



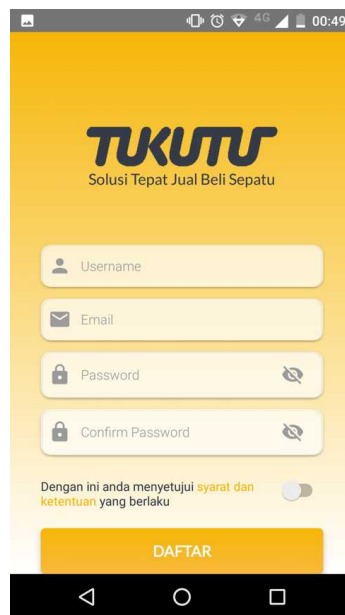
Gambar 4.18: Halaman Login

4.6.2 Halaman Login Provider



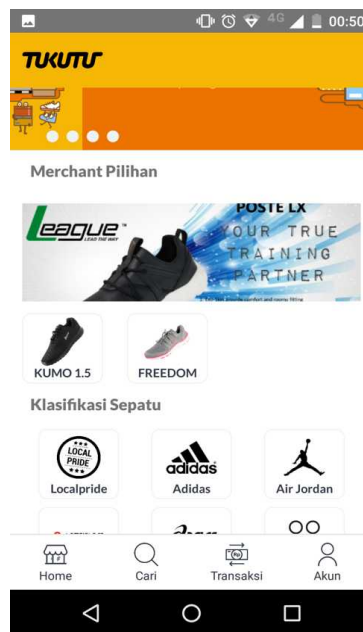
Gambar 4.19: Halaman Login Provider

4.6.3 Halaman Register



Gambar 4.20: Halaman Register

4.6.4 Halaman Utama



Gambar 4.21: Halaman Utama

4.6.5 Halaman Profil



Gambar 4.22: Halaman Profil

4.6.6 Halaman Daftar Transaksi



Gambar 4.23: Halaman Daftar Transaksi

4.6.7 Halaman Gabung Jadi Penjual

The screenshot shows a mobile application interface for joining as a seller. At the top, there is a back arrow and a status bar with the time 00:51. The form contains several input fields: 'Masukkan Nama merchant', 'Masukkan nomor telepon', 'Masukkan nama pemilik rekening', 'Masukkan nama Bank', 'Masukkan nomer rekening', 'Kota Merchant', and 'Masukkan alamat lengkap merchant'. Below these fields is a section for 'Upload KTP' with a 'Browse' button and a 'SIMPAN' (Save) button at the bottom.

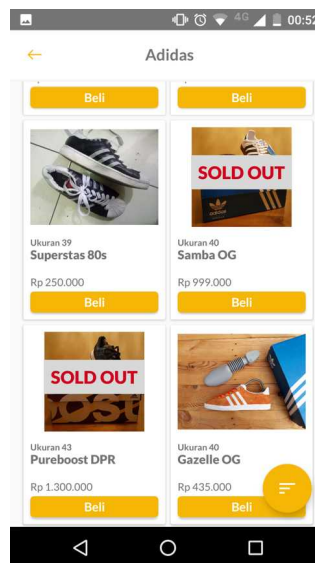
Gambar 4.24: Halaman Gabung Jadi Penjual

4.6.8 Halaman Manajemen Penjual

The screenshot shows a mobile application interface for seller management. At the top, there is a back arrow and a status bar with the time 02:47. The dashboard features a profile section with a circular profile picture, the name 'Tukutu Indonesia', email 'id.tukutu@gmail.com', and phone number '089623040007'. Below this is an 'Upload Sepatu' (Upload Shoes) section with three icons: 'M-Sepatu', 'M-Transaksi', and 'M-Pesan'. At the bottom, there is a 'Kelola Diskusi' (Manage Discussion) section with a 'Panduan Merchant' (Merchant Guide) link.

Gambar 4.25: Halaman Manajemen Penjual

4.6.9 Halaman Daftar Sepatu



Gambar 4.26: Halaman Daftar Sepatu

4.6.10 Halaman Pembelian



Gambar 4.27: Halaman Pembelian

4.6.11 Halaman Konfirmasi Pembelian

Detail Pesanan
Authentic Navy Blue
Ukuran 42.5 Kondisi used 90 %

Detail Pengiriman
Husein Indra Kusuma
Jalan Abimanyu 5 47 Pendrikan Lor Semarang Tengah
085788881425
Citra Van Titipan Kilat (TIKI)

Detail Pembayaran
Harga Barang Rp 150.000
Harga Ongkir Rp 57.000
Harga Potongan Rp 0
Harga Total Rp 207.000

Kode Voucher
Kode Voucher

Gambar 4.28: Halaman Konfirmasi Pembelian

4.6.12 Halaman Pembayaran

Detail Pesanan
Authentic Navy Blue
Rp 207.000

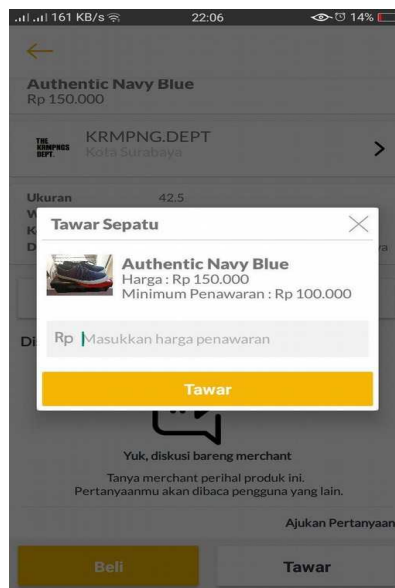
Rekening TUKUTU
BCA a/n Husein Indra Kusuma
009-1725-451

Bukti Transfer
Foto bukti pembayaran

☐ Saya setuju dengan syarat dan ketentuan yang berlaku

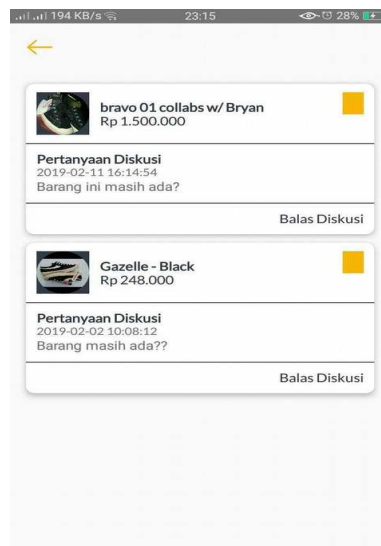
Gambar 4.29: Halaman Pembayaran

4.6.13 Halaman Penawaran



Gambar 4.30: Halaman Tawar

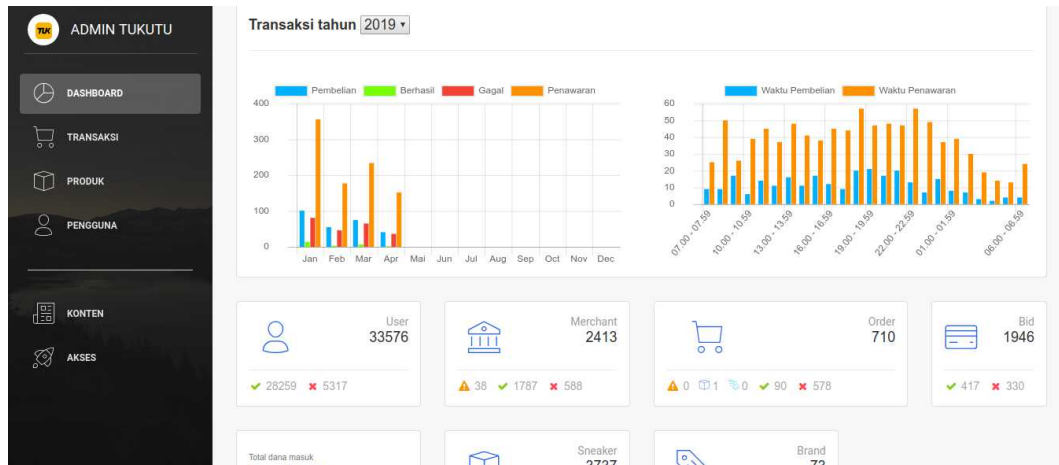
4.6.14 Halaman Diskusi



Gambar 4.31: Halaman Daftar Diskusi

4.7 Implementasi Dashboard Admin (Mimin)

4.7.1 Halaman Utama



Gambar 4.32: Halaman Utama Admin

4.7.2 Halaman Transaksi

ID IT	BARANG IT	PENJUAL IT	PEMBELI IT	WAKTU PEMBELIAN IT	STATUS IT
16	Losing Grip: casual	Fitri nuraeni	iqbal ramadhan	5 bulan yang lalu	Sudah diterima
42	Vans: Old Skool Gum Sole	Nusa_store	Muhammad Hafizh Muzaky	5 bulan yang lalu	Sudah diterima
55	Nike: Vapormax Triple Black 3.0	KayGoods	albio tanviano	5 bulan yang lalu	Sudah diterima
59	Adidas: Continental 80	SOLESAL STORE	muhammad faiz firmanda	5 bulan yang lalu	Sudah diterima
64	Converse: Converse CT All Star Ox Double Tongue Black/Red [W867]	BLESSINGFOOT	Muhammad Panthera Ardaffaka	5 bulan yang lalu	Sudah diterima
82	Converse: Chuck Taylor	Second Label	Fahmi Kuncoro	5 bulan yang lalu	Sudah diterima
97	Converse: CHUK TAYLOR II HI CAMO REFLECTIVE	RESIGNMENT.ID	Faris Luthfi Kurniawan	5 bulan yang lalu	Sudah diterima
98	Vans: Vault Slip-On OG True White Canvas	Hannovercure	made nranatavasa	5 bulan yang lalu	Sudah diterima

Gambar 4.33: Halaman Transaksi

4.7.3 Halaman Produk

TUK

ADMIN TUKUTU

DASHBOARD

TRANSAKSI

PRODUK

PENGGUNA

KONTEN

AKSES

Sepatu

Brand

TOTAL : 3737

Cari

10

Reload

ID	MERCHANT	BRAND	TIPE	UKURAN	WARNA	HARGA
4	wafflesfootwear	Waffles	SNAKE toddler slip on	22	green/black	RP 399.000
5	pairshoes Malang	Vans	SK8hi Tumble	42	white	RP 1.199.000
6	Monkeysquad	Vans	Oldskool OG LX Forest Night	42	Hijau	RP 1.850.000
7	ZEEGOODS	Vans	Slip On Classic Checkerboard	40.5	Hitam / Putih	RP 799.000
8	Rowley Store	Vans	Old Skool V Side Stripe Black	9	Hitam	RP 1.200.000
9	Hayes Footwear	Hayes	Phantom Black White	39	Hitam dan Putih	RP 300.000
10	Betenstore.id	Vans	Vans Era Pro Anniversary 50th Stripe Checkerboard	40.5	Stripes Checkerboard	RP 1.100.000
11	BLESSINGFOOT	Vans	Vans Slip On Og Vault True White	9	Putih	RP 850.000
12	salvok store official	Vans	old skool	39	white green jade	RP 1.300.000
13	BLESSINGFOOT	Vans	Vans Slip On Og Vault True White	9	Putih	RP 850.000

Gambar 4.34: Halaman Produk

4.7.4 Halaman Pengguna

TUKU

ADMIN TUKUTU

DASHBOARD

TRANSAKSI

PRODUK

PENGGUNA

KONTEN

AKSES

User

Merchant

Merchant +

Merchant Banned

Tukutu Official

TOTAL : 38

Cari

10

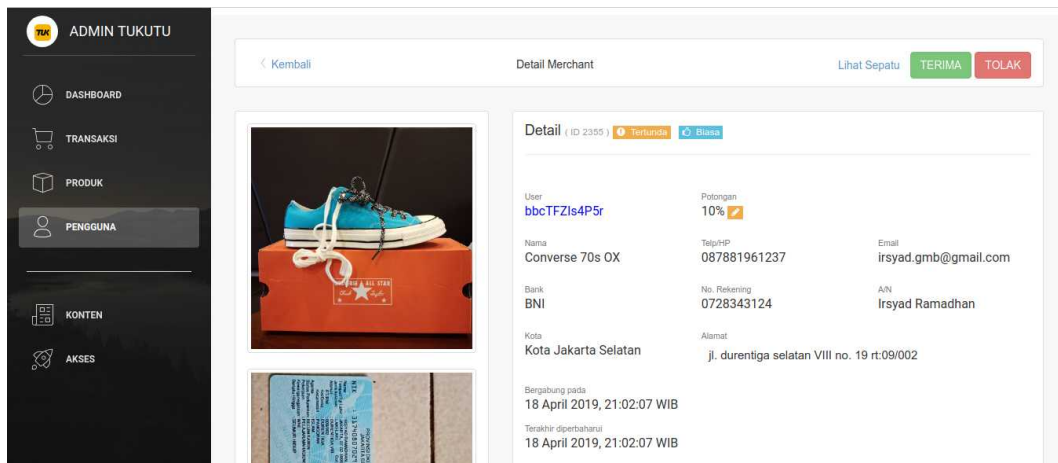
Tertunda

Reload

ID	NAMA	KOTA	TEL/PHP	EMAIL	STATUS	WAKTU BERGABUNG
2355	Converse 70s OX	Kota Jakarta Selatan	087881961237	irsyad.gmb@gmail.com	Tertunda	6 hari yang lalu
2357	Bandot Shop	Kota Serang	081247951285	starfox212222@gmail.com	Tertunda	6 hari yang lalu
2358	threel sneakers	Kota Yogyakarta	087825485522	ardianurokman@gmail.com	Tertunda	6 hari yang lalu
2359	MixedUp.Hype	Kota Semarang	081315157488	stephs.detailshop@gmail.com	Tertunda	6 hari yang lalu
2363	Kelvin Riandy	Kabupaten Bogor	+6287832753083	kelvinriandy21@gmail.com	Tertunda	5 hari yang lalu
2366	kid_flip	Kota Jakarta Timur	081234345666	firlyflip@gmail.com	Tertunda	5 hari yang lalu
2367	wishlist	Kabupaten Karanganyar	085336442860	lowstore.official@gmail.com	Tertunda	5 hari yang lalu
2369	Jefri Store	Kota Surabaya	083847732722	rm.jefri@gmail.com	Tertunda	4 hari yang lalu
2373	prelovedck	Kota Bandung	08112069100	tropicalvibes@gmail.com	Tertunda	3 hari yang lalu
2376	DealBuy	Kota Pekalongan	089619446735	fauzanspenda@gmail.com	Tertunda	3 hari yang lalu

Gambar 4.35: Halaman Pengguna

4.7.5 Halaman Seleksi Penjual



Gambar 4.36: Halaman Seleski Penjual

4.8 Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi yang telah dibangun dapat berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan.

4.8.1 White Box Testing

Pada pengujian *white box* disini akan menggunakan teknik *Basic Path*. Pada *Basic path*, dilakukan pengujian dengan membuat test case yang didapatkan dari jalur indepeden. Jalur independen tersebut didapatkan dari menghitung nilai *cyclomatic complexity* dari flowgraph dengan rumus berikut:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* pada grafik

N = jumlah *node* pada grafik

Untuk melakukan pengujian *white box* digunakan fungsi utama yaitu order (pembelian).

```
if ($invalid = $this->response->validate($req, Rule::ORDER)) return $invalid;
try {
    $sneaker = Sneaker::findOrFail($sneaker_id);

    // SIZE SNEAKER
    if ($req->filled('size_id')) {
        $size = $sneaker->sizes()->find($req->size_id);
        if (!$size) return $this->response->messageError('Size not found', 404);
    }
    else $size = $sneaker->sizes()->first();

    // VALIDATE CONDITION SNEAKER
    if ($notValid = $this->validateOrder($req, $sneaker)) return $notValid;

    // IF STOCK = 1. THE SNEAKER HAS RECENTLY BID, CANCEL ALL BID
    if ($sneaker->stock == 1) $this->isStockIsOne($sneaker);

    // START cost calculation
    $price = $sneaker->price * $req->count;
    $fee_tukutu = ceil(round($price * $sneaker->merchant->cut_fee, 2));
    $price_merchant = $price - $fee_tukutu;
    $price_transaction = $price + $req->courier_cost;
```

Gambar 4.37: Source Code Order bagian a

```
$req->merge([
    'user_id' => $this->user->id,
    'sneaker_id' => $sneaker->id,
    'sneaker_price' => $sneaker->price,
    'sneaker_size' => $size->size,
    'sneaker_color' => $sneaker->color,
    'price' => $price_transaction,
    'price_merchant' => $price_merchant,
    'size_id' => $size->id
]);
// END cost calculation

$order = OrderModel::create($req->all());
$order->transaction()->create($req->only('price', 'price_merchant', 'count'));
$sneaker->decrement('stock', $req->count);
$size->decrement('stock', $req->count);
```

Gambar 4.38: Source Code Order bagian b

```
// VOUCHER USAGE
if ($req->filled('code')) {
    $voucher = $this->validateOrderVoucher($req->code, $price);

    if ($voucher->valid) {
        $usageVoucher = VoucherUsage::create([
            'user_id' => $this->user->id,
            'voucher_id' => $voucher->id,
            'order_id' => $order->id
        ]);

        if ($usageVoucher->voucher->has_limit) $usageVoucher->voucher()->decrement('limit');
        if (!$voucher->is_free_ship) {
            $price = $price - (($voucher->is_percent) ? (int)((($voucher->value/100) * $price):$voucher->value) + $req->courier_cost;
        }
        $order->transaction()->update(['price' => $price]);
    }

    $order->voucher_usage = $voucher;
}
```

Gambar 4.39: Source Code Order bagian c

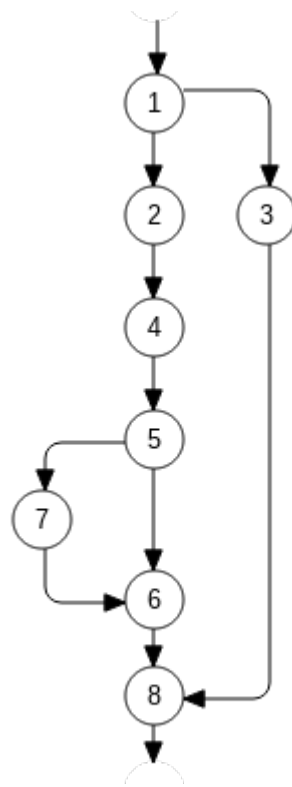
```
// Auto Cancel Call
$this->callAutoCancel($order->transaction->id);

LogTukutu::create($req, $this->user, ['activity', 'process', 'order sneaker id:'.$sneaker_id, ['order' => $order, 'sneaker' => $sneaker]])
return $this->send($order);
} catch (Exception $e) {
    if ($e instanceof \Illuminate\Database\Eloquent\ModelNotFoundException)
        return $this->response->messageError('Sneaker not found', 404);

    return $this->response->serverError();
}
```

Gambar 4.40: Source Code Order bagian d

Dari source coder tersebut kemudian dibuat flowgraph untuk menghitung nilai *cyclomatic complexity* dan menentukan jalur,



Gambar 4.41: Flowgraph Order (pembelian)

Dari folwgraph kemudian menghitung *cyclomatic complexity* dengan rumus sebelumnya $V(G) = E - N + 2$.

$$V(G) = E - N + 2$$

$$V(G) = 9 - 8 + 2$$

$$V(G) = 3$$

Dari hasil perhitungan *cyclomatic complexity* dibuat jalur independen yang diketahui memiliki 3 jalur.

Jalur 1 = 1 – 3 – 8

Keterangan : Terjadi pembelian tapi gagal karena data yang dikirim tidak valid atau syarat pembelian tidak terpenuhi.

Jalur 2 = 1 – 2 – 4 – 5 – 6 – 8

Keterangan : Terjadi pembelian dan berhasil.

Jalur 3 = 1 – 2 – 4 – 5 – 7 – 6 – 8

Keterangan : Terjadi pembelian yang menggunakan voucher dan berhasil.

Hasil pengujian test case dapat dilihat ditabel berikut:

Tabel 4.1: Test Case White Box

No	Test Case	Ketercapaian
1	1 – 3 – 8	Berhasil
2	1 – 2 – 4 – 5 – 6 – 8	Berhasil
3	1 – 2 – 4 – 5 – 7 – 6 – 8	Berhasil

4.8.2 Black Box Testing

Pengujian *Black Box* adalah suatu cara yang digunakan untuk menguji spesifikasi fungsional dari perangkat lunak (*software*). Pada pengujian *Black Box* menggunakan Postman untuk mengetahui hasil response jika berhasil, pengujian dinyatakan berhasil jika objek *success* bernilai *true*. Hasil pengujian dibuat dalam tabel berikut:

Tabel 4.2: Black Box Testing

No	Proses	Hasil yang diharapkan	Hasil
1	Register	status : - success : true	Berhasil

		- code : 201 - message : “user has been created, please verified your account” result : null	
2	Login	status : - success : true - code : 200 - message : “OK” result : - token	Berhasil
3	Login Provider	status : - success : true - code : 200 - message : “OK” result : - token	Berhasil
4	Order	status : - success : true - code : 200 - message : “OK” result : - data transaksi	Berhasil
5	Bid	status : - success : true - code : 201 - message : “new bid added” result : null	Berhasil
6	Diskusi	status : - success : true	Berhasil

		- code : 200 - message : “OK” result : - data diskusi	
7	Balas Diskusi	status : - success : true - code : 200 - message : “OK” result : - data diskusi	Berhasil