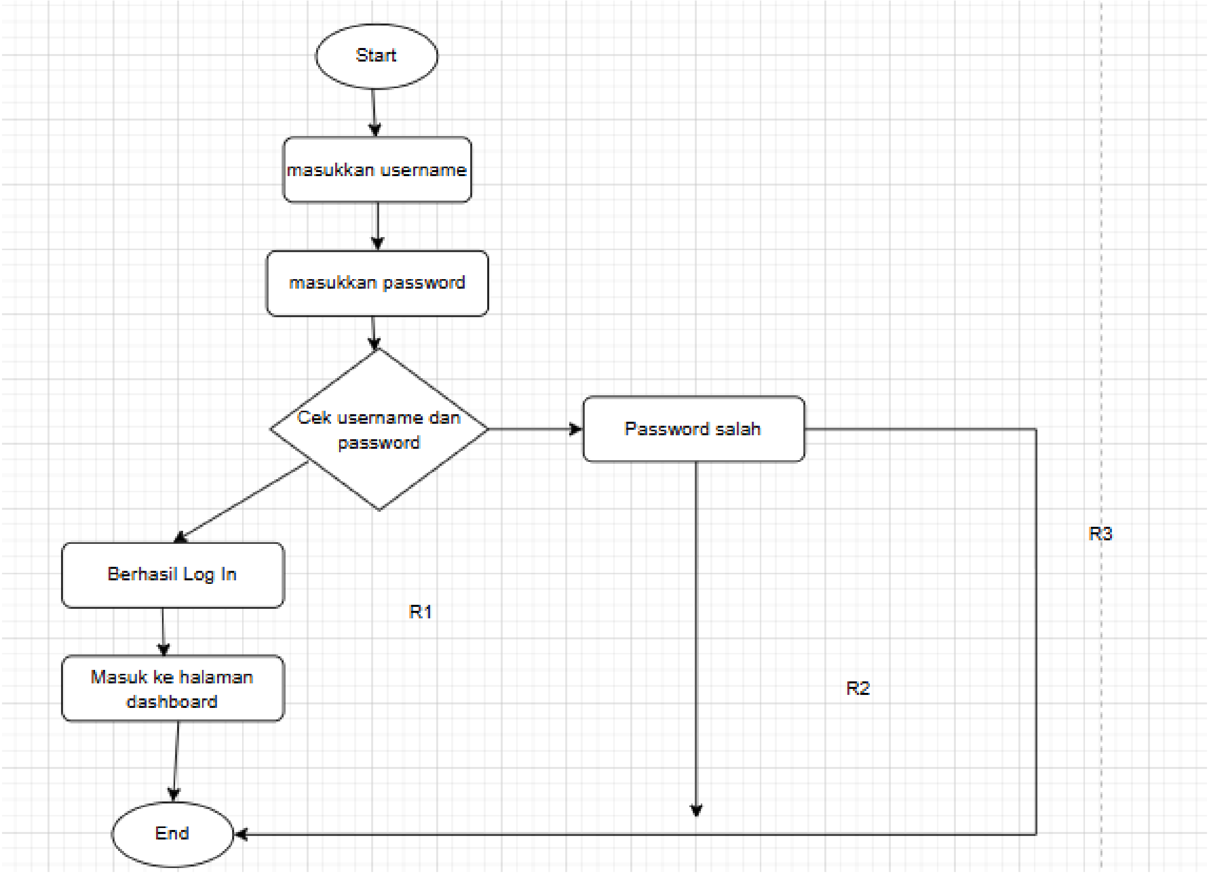
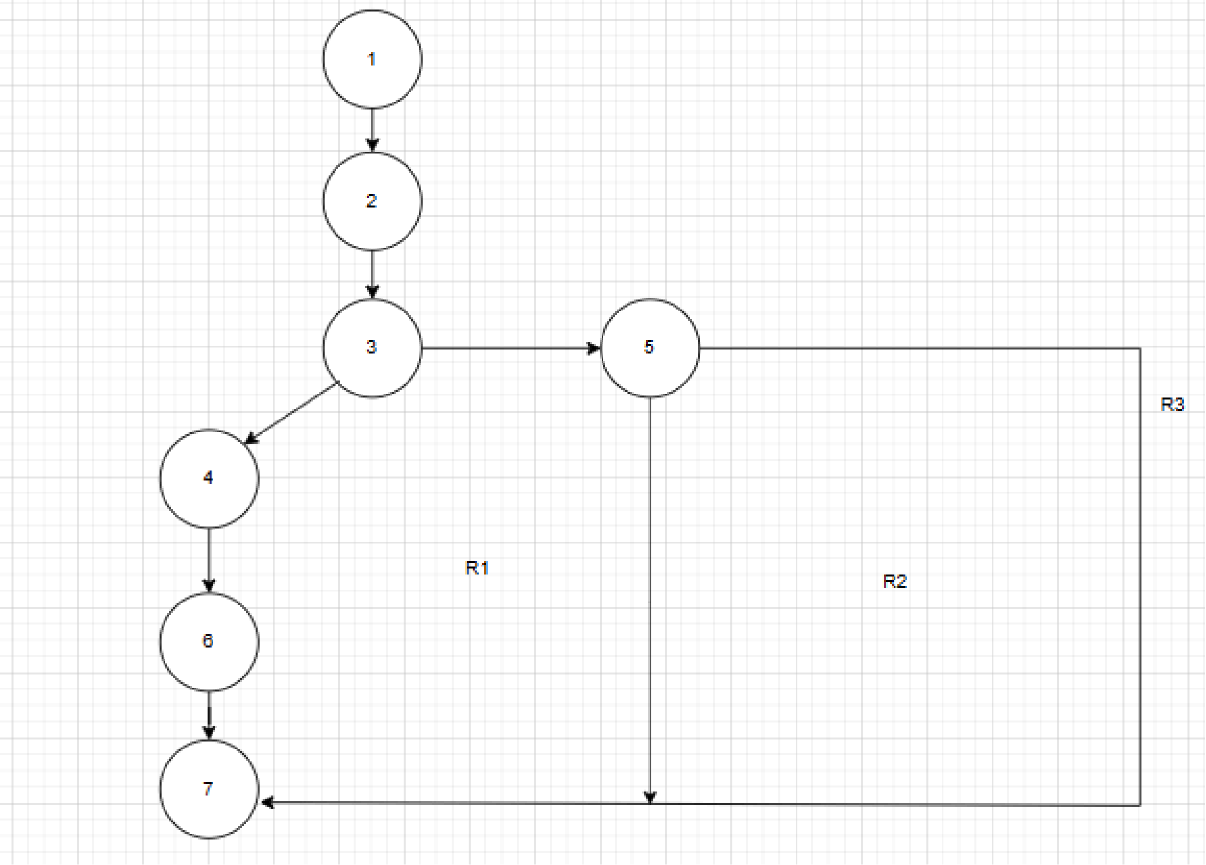
Pengujian White Box Project Toko Sunali Jaya

Login Admin

```
@app.route('/admin/login', methods=['GET', 'POST'])
def admin_login():
if request.method == 'POST':
username = request.form['username']
password = request.form['password']
admin = Admin.query.filter_by(username=username).first()
if admin and check_password_hash(admin.password, password):
session['admin'] = admin.username
return redirect(url_for('admin_dashboard'))
else:
return render_template('admin_login.html', error="Invalid credentials!")
return render_template('admin_login.html')
```

Flowchart & FlowGraph Login





Region	**:**	**3**
Edge	**:**	**8**
Node	**:**	**7**
Simpul Predikat	**:**	**3**
Independent path	**:**	**3**
		1-2-7
		1-3-4-7
		1-5-6-7

Kompleksitas Siklomatis **: V(G) = E - N + 2**

8 - 7 + 2 = 3

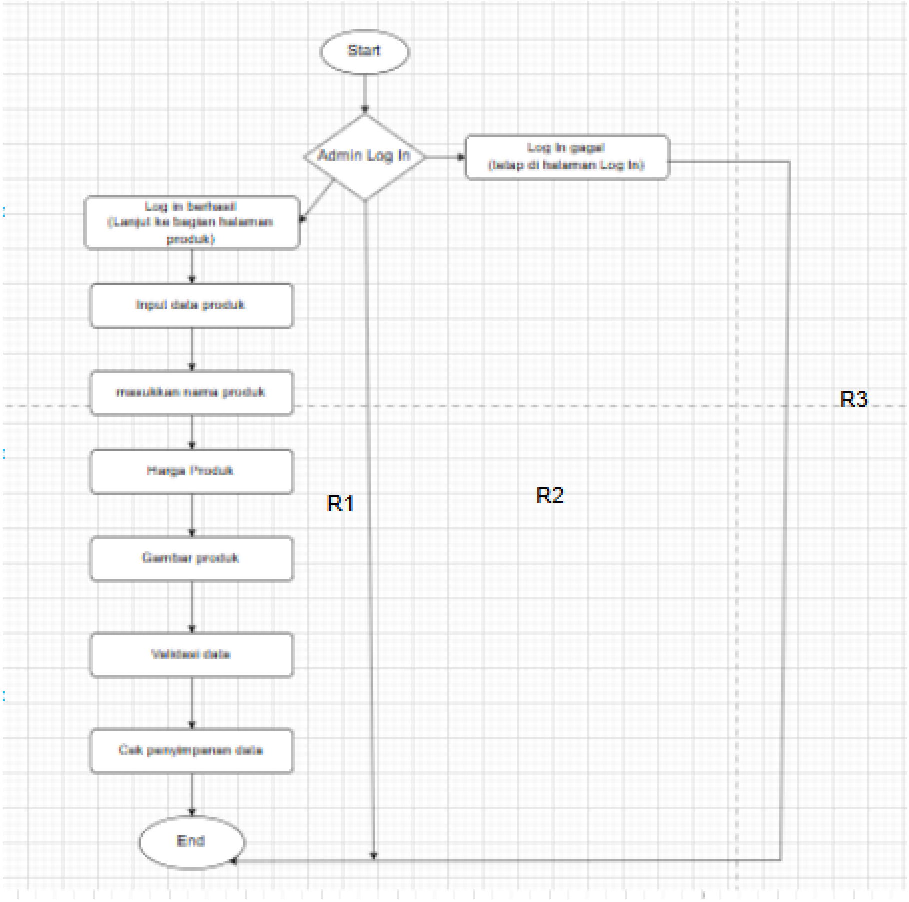
V(G) = P + 1

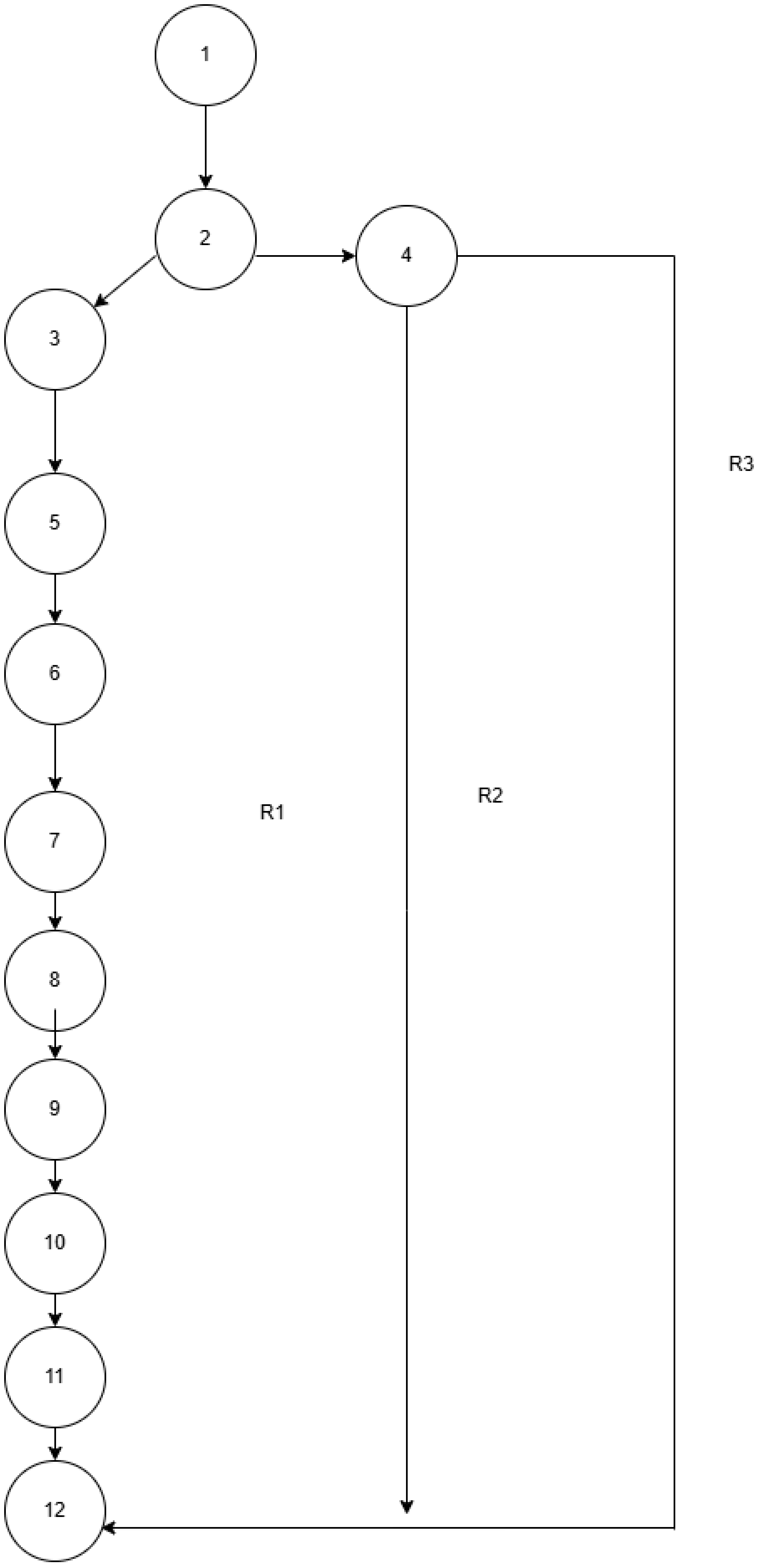
2 + 1 = 3

masukkan produk ke database dari admin

```
@app.route('/admin/add', methods=['GET', 'POST'])
def add_product():
    if 'admin' not in session:
        return redirect(url_for('admin_login'))

    if request.method == 'POST':
        name = request.form['name']
        price = float(request.form['price'])
        description = request.form['description']
        image = request.files['image']

        if image and allowed_file(image.filename):
            filename = secure_filename(image.filename)
            filepath =
os.path.join(app.config['UPLOAD_FOLDER'], filename)
            image.save(filepath)

            new_product = Products(name=name, price=price,
description=description, image=filename)
            db.session.add(new_product)
            db.session.commit()

        return redirect(url_for('admin_dashboard'))

    return render_template('add_product.html')
```

Flowchart & FlowGraph masukkan produk ke database dari admin




1
2
4
3
5
6
7
8
9
10
11
12
R1
R2
R3

N = 12 Simpul Predikat = 2

E = 13 Independent Path = 3 (1- 2-3-12), (1-4-5-6-12), (1-4-7-8-12)

R = 3

Kompleksitas Siklomatis = V(G) = E - N + 2

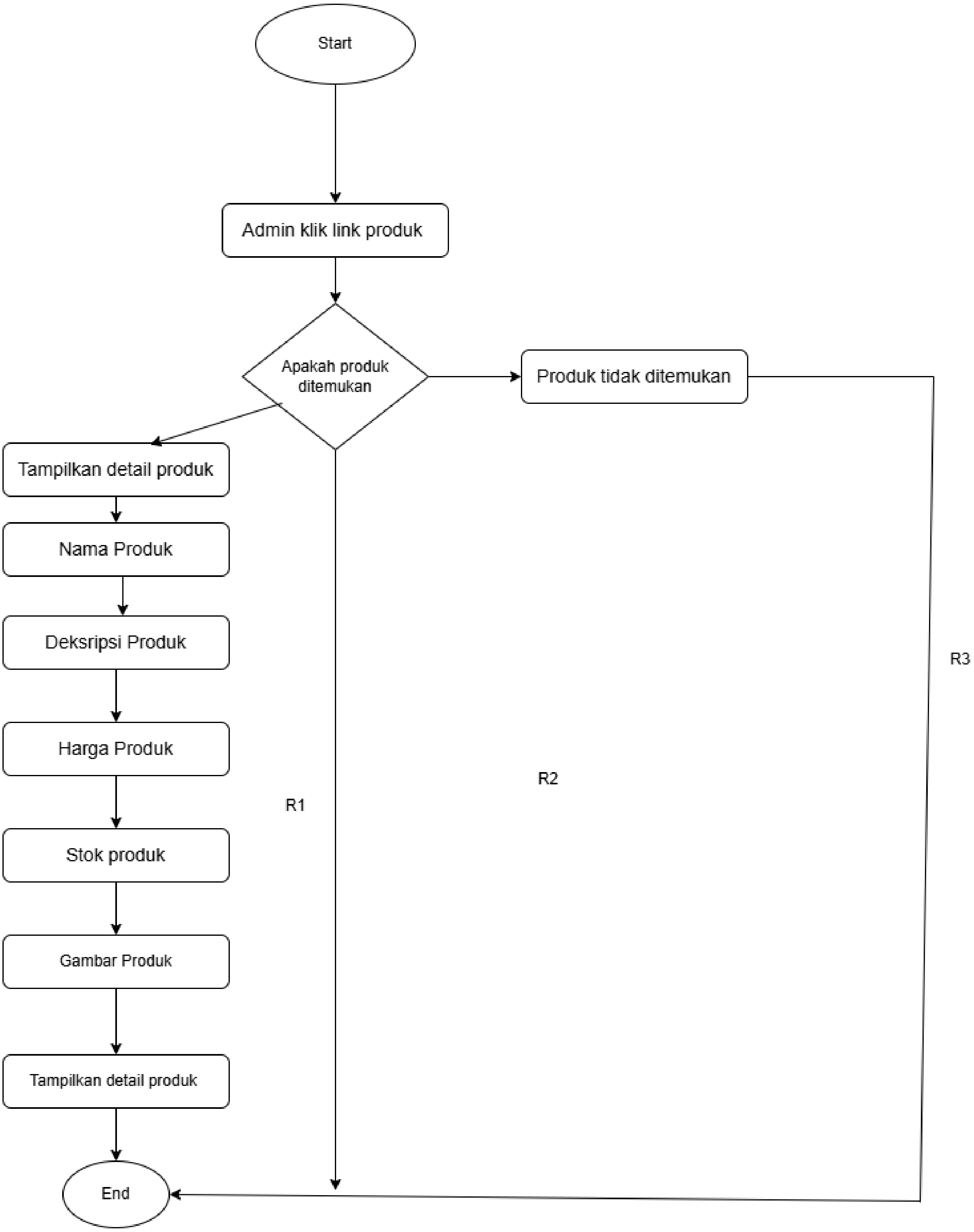
13 - 12 + 2 = 3

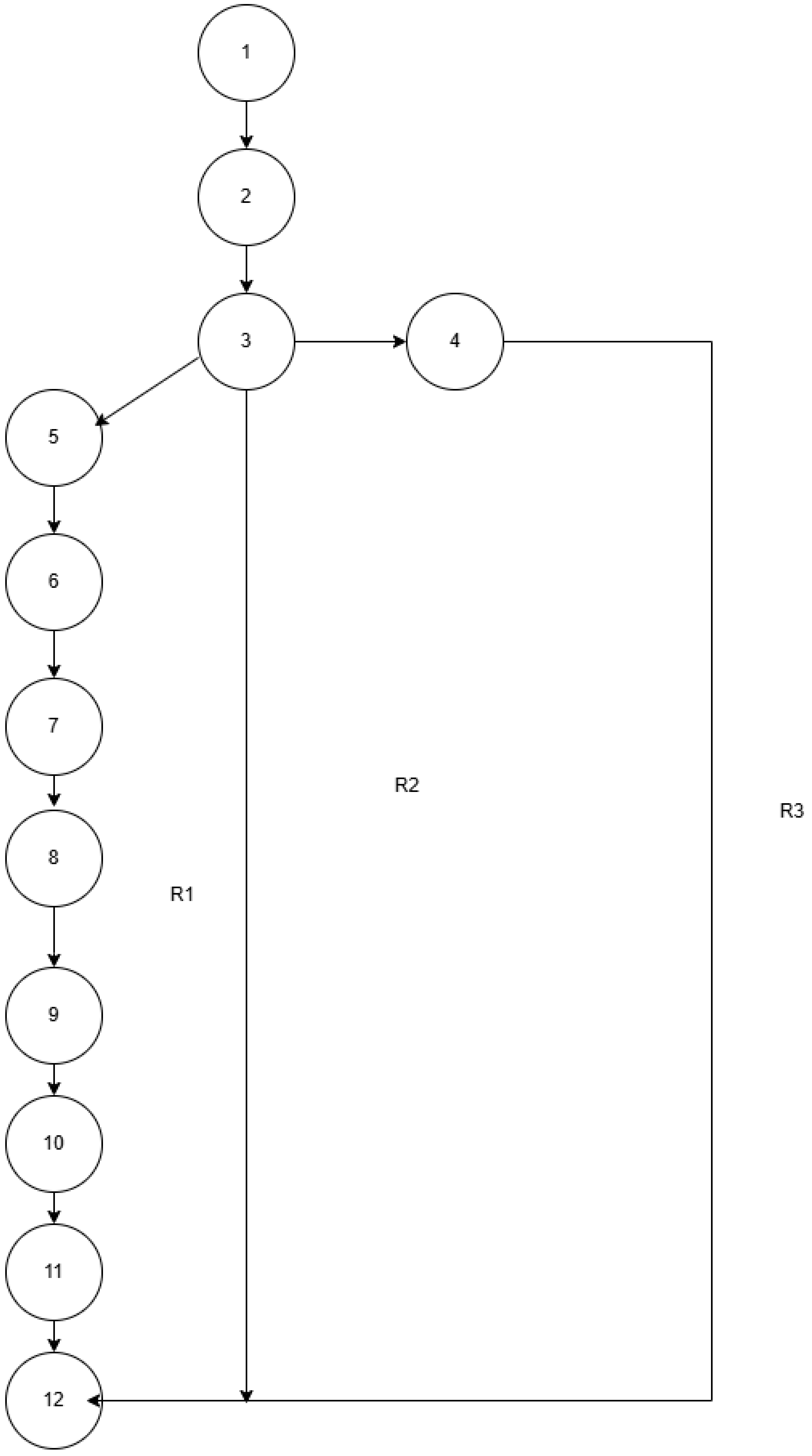
= V(G) = P + 1

2 + 1 = 3

```
#Bagian detil produk
@app.route('/product/<int:product_id>')
def product_detail(product_id):
product = Products.query.get_or_404(product_id)
return render_template('product_detail.html',
product=product)
```

Flowchart dan Flowgraph Bagian detail produk


Start
Admin klik link produk
Apakah produk ditemukan
Produk tidak ditemukan
Tampilkan detail produk
Nama Produk
Deksripsi Produk
Harga Produk
Stok produk
Gambar Produk
Tampilkan detail produk
End
R1
R2
R3


1
2
3
4
5
6
7
8
9
10
11
12
R1
R2
R3

N = 12 Simpul Predikat = 2

E = 13 Independent Path = 3 (1-2-3-12), (1-4-5-6-12), (1-4-7-8-12)

R = 3

Kompleksitas Siklomatis = V(G) = E - N + 2

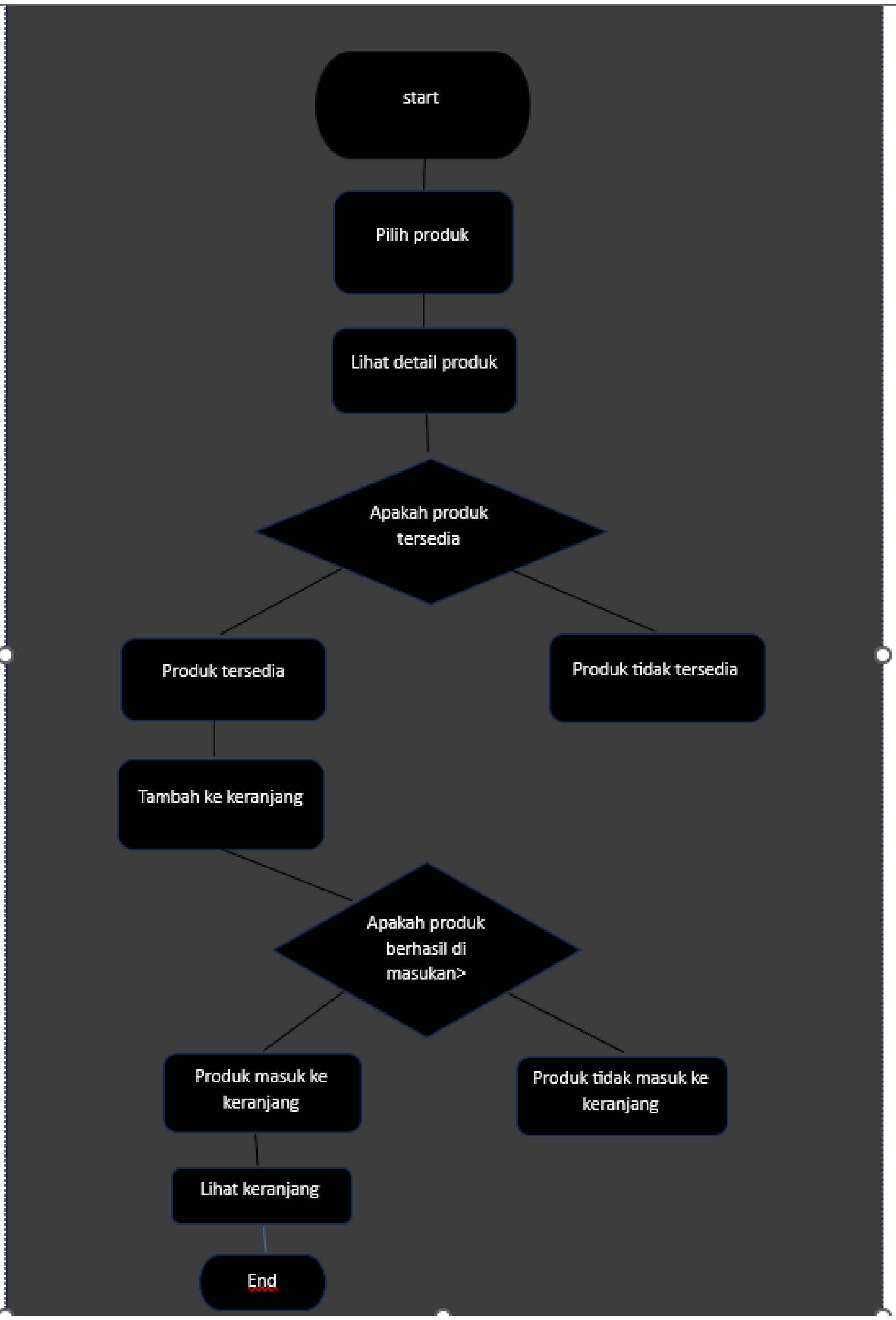
3 - 12 + 2 = 3

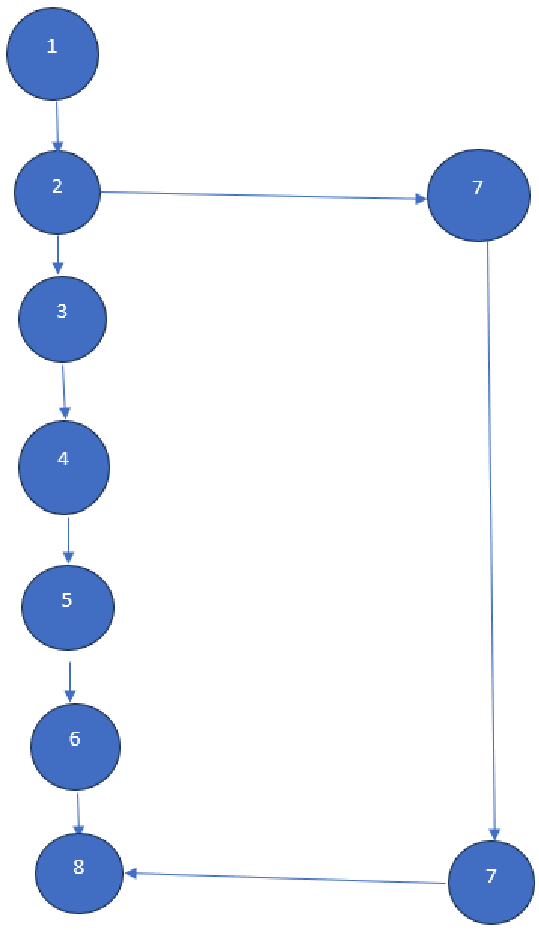
= V(G) = P + 1

2 + 1

payment-complete

```
#Pembayaran selesai
@app.route('/payment_complete', methods=['GET', 'POST'])
def payment_complete():
file = request.files.get('payment_proof')
if file and allowed_file(file.filename):
filename = secure_filename(file.filename)
filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
file.save(filepath)
order_id = session.get('order_id')
order = Order.query.get(order_id)
if order:
order.payment_proof = filename
order.payment_status = 'Menunggu Verifikasi'

db.session.commit()

flash("Bukti pembayaran berhasil diunggah dan pesanan diperbarui!", "success")

else:

flash("Pesanan tidak ditemukan.", "danger")

else:

flash("Harap unggah file yang valid.", "danger")

session.pop('cart', None)

session.pop('checkout_complete', None)

session.pop('payment_method', None)

return render_template('payment_complete.html')
```


start
Pilih produk
Lihat detail produk
Apakah produk tersedia
Produk tersedia
Produk tidak tersedia
Tambah ke keranjang
Apakah produk berhasil di masukan>
Produk masuk ke keranjang
Produk tidak masuk ke keranjang
Lihat keranjang
End



E = 11

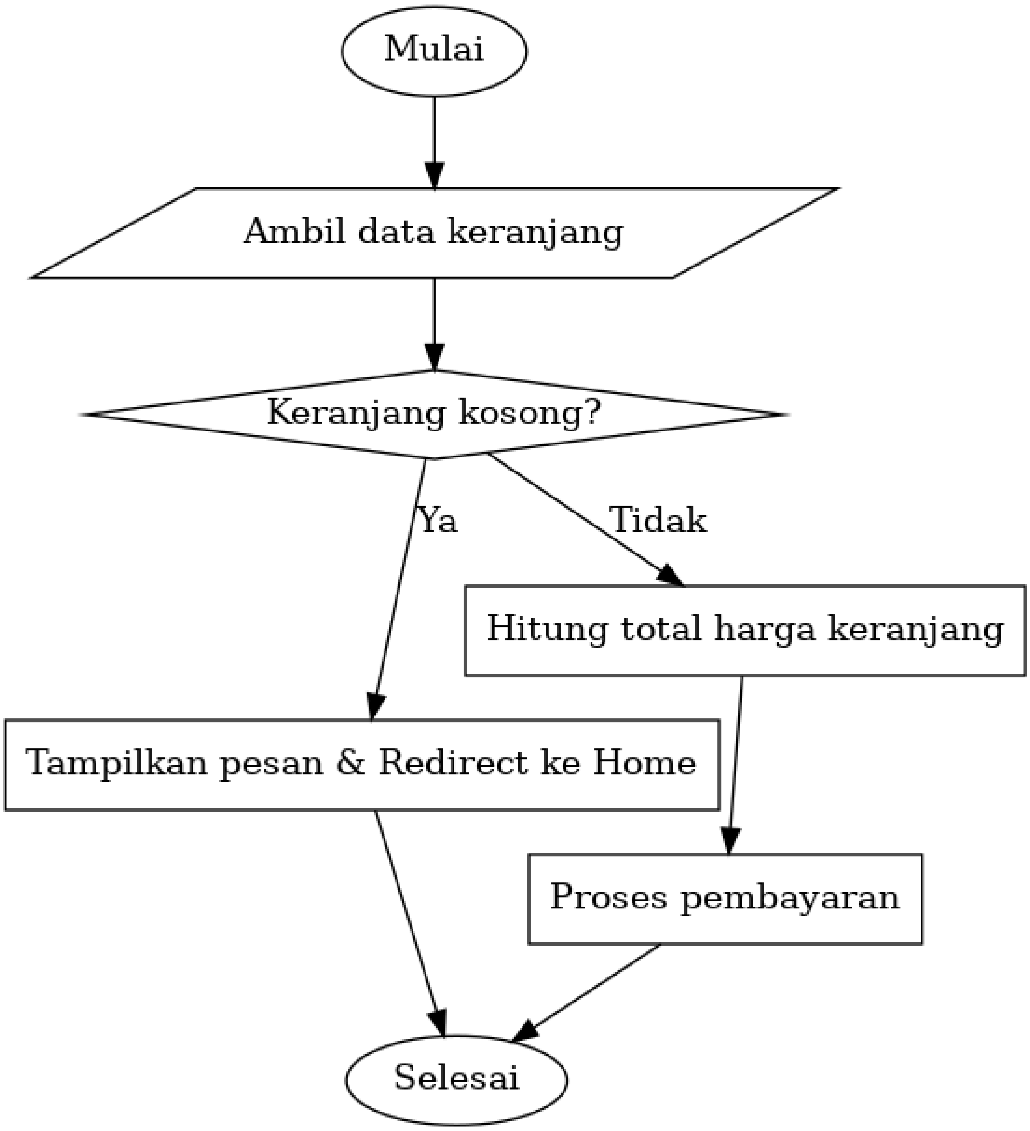
N = 9

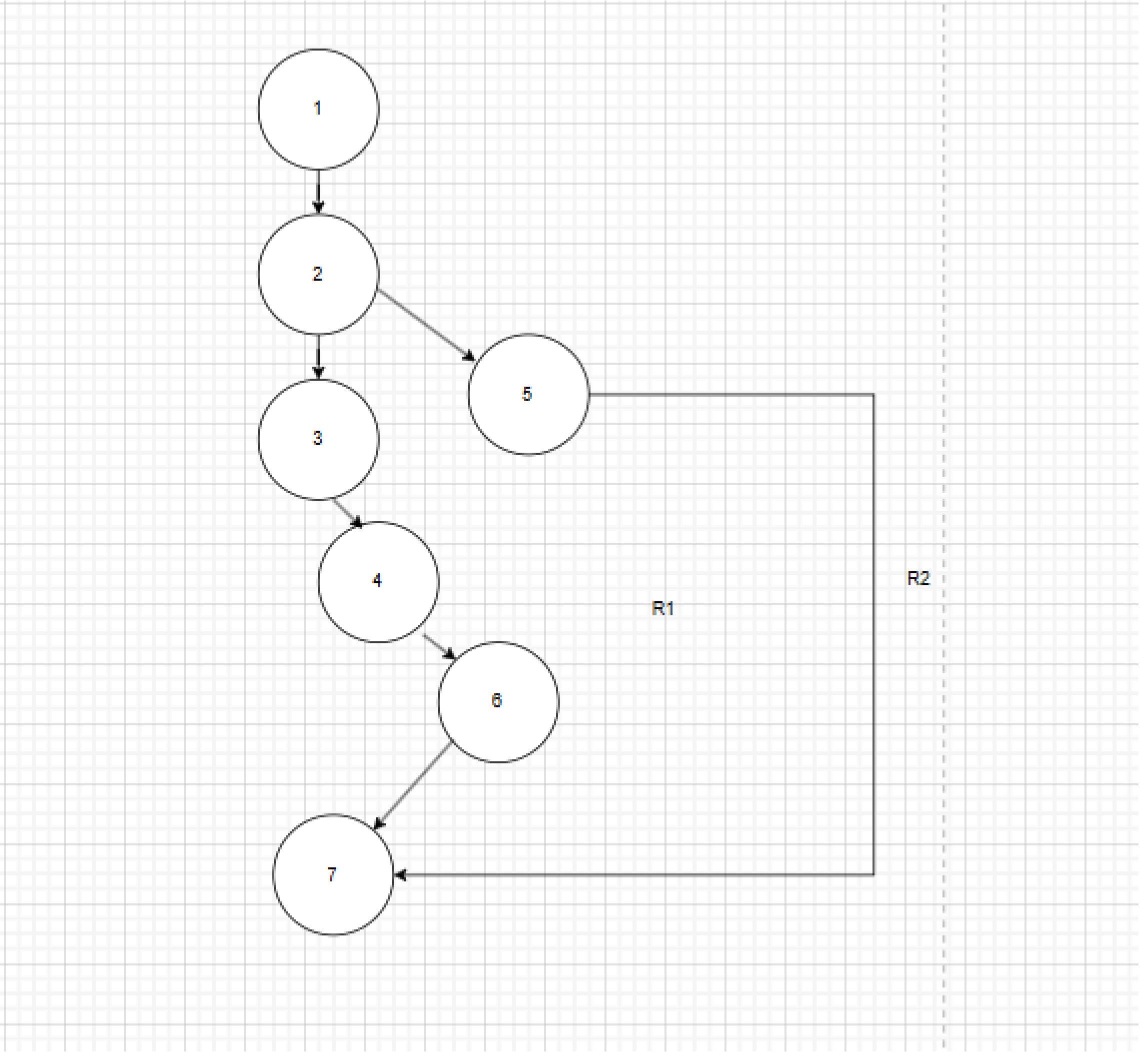
P = 1

Kompleksitas siklomatis = M = 11 - 9 + 2 * 1 = 4

Cart

```
#Bagian pembayaran
@app.route('/payment', methods=['GET', 'POST'])
def payment():
cart = session.get('cart', [])
if not cart:
flash("Keranjang Anda kosong. Silakan tambah produk terlebih dahulu.", "warning")
return redirect(url_for('home'))
cart_total = sum(item['price'] * item['quantity'] for item in cart)
```


Mulai
Ambil data keranjang
Keranjang kosong?
Ya
Tidak
Hitung total harga keranjang
Tampilkan pesan & Redirect ke Home
Proses pembayaran
Selesai



N = 12 Simpul Predikat = 2

E = 13 Independent Path = 3 (1-2-3-12), (1-4-5-6-12), (1-4-7-8-12)

R = 3

Kompleksitas Siklomatis = V(G) = E - N + 2

3 - 12 + 2 = 3

= V(G) = P + 1

2 + 1

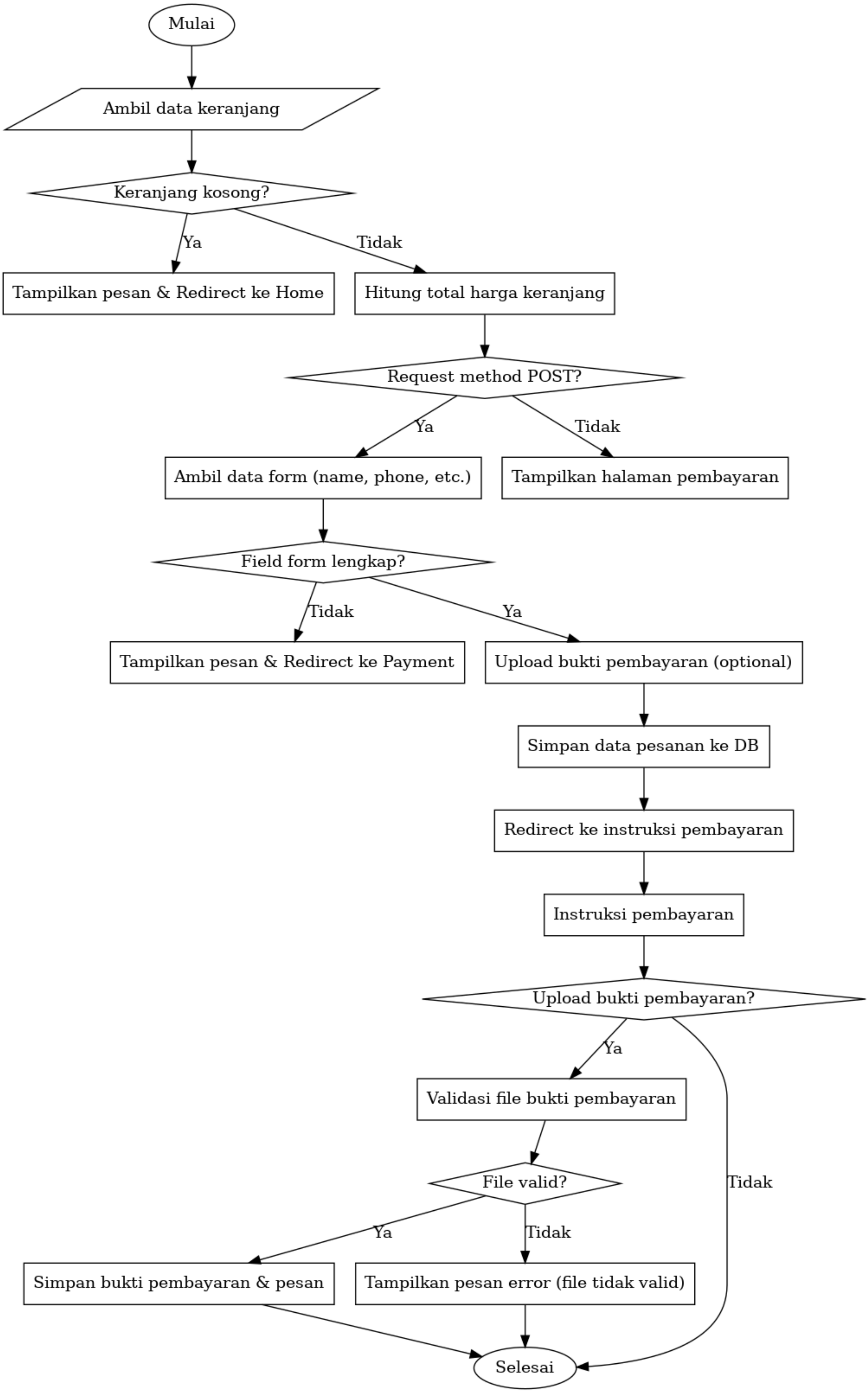
Halaman Pembayaran dan Pengolahan Pesanan

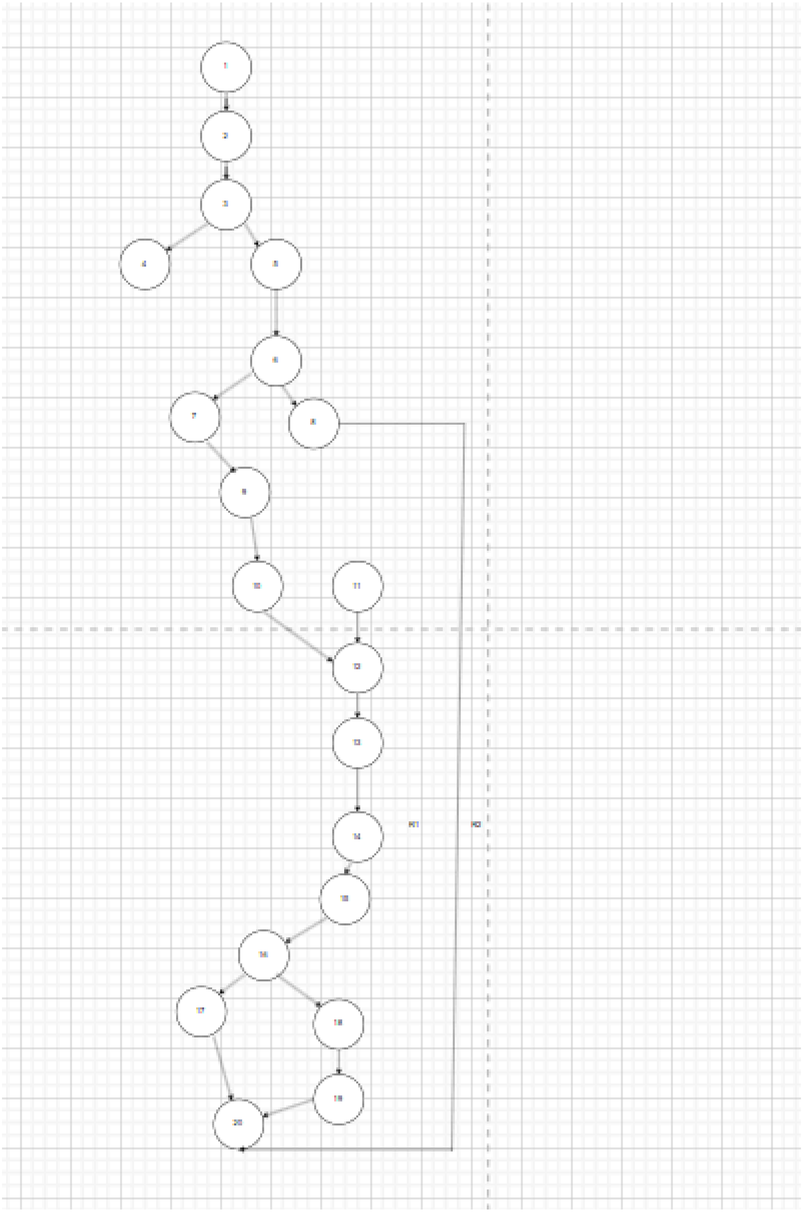
```
@app.route('/payment', methods=['GET', 'POST'])
def payment():
cart = session.get('cart', [])
if not cart:
flash("Keranjang Anda kosong. Silakan tambah produk terlebih dahulu.", "warning")
return redirect(url_for('home'))
cart_total = sum(item['price'] * item['quantity'] for item in cart)
if request.method == 'POST':
name = request.form.get('name')
phone = request.form.get('phone')
address = request.form.get('address')
payment_method = request.form.get('payment_method')
if not name or not phone or not address or not payment_method:
flash("Semua field wajib diisi.", "warning")
return redirect(url_for('payment'))
payment_proof = None
file = request.files.get('payment_proof')
if file and allowed_file(file.filename):
filename = secure_filename(file.filename)
file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
payment_proof = filename
new_order = Order(
customer_name=name,
customer_phone=phone,
customer_address=address,
total_price=cart_total,
payment_method=payment_method,
payment_status='Pending',
payment_proof=payment_proof,
cart_data=cart
)
db.session.add(new_order)
db.session.commit()
if payment_method:
flash(f"Pesanan Anda telah diterima. Silakan lanjutkan dengan {payment_method}.", "success")
return redirect(url_for('payment_instructions', method=payment_method))
else:
flash("Silakan lengkapi detail tambahan untuk pembayaran.", "info")
return redirect(url_for('payment_form'))
return render_template('payment.html', cart=cart, cart_total=cart_total)
```

instruksi pembayaran berdasarkan metode pembayaran yang dipilih oleh pengguna

```
@app.route('/payment-instructions/<method>', methods=['GET', 'POST'])
def payment_instructions(method):
cart = session.get('cart', [])
cart_total = sum(item['price'] * item['quantity'] for item in cart)
```

```
if request.method == 'POST':

file = request.files.get('payment_proof')

if file:

if allowed_file(file.filename):

filename = secure_filename(file.filename)

filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)

file.save(filepath)

session['payment_proof'] = filename

flash("Bukti pembayaran berhasil diupload!", "success")

return redirect(url_for('payment_instructions', method=method))

else:

flash("Tipe file tidak diperbolehkan. Silakan upload gambar dengan format yang valid.", "danger")

else:

flash("Silakan pilih gambar bukti pembayaran untuk diupload.", "warning")

return render_template(f'instructions/{method}.html', method=method, cart_total=cart_total)
```


Mulai
Ambil data keranjang
Keranjang kosong?
Ya
Tidak
Tampilkan pesan & Redirect ke Home
Hitung total harga keranjang
Request method POST?
Ya
Tidak
Ambil data form (name, phone, etc.)
Tampilkan halaman pembayaran
Field form lengkap?
Tidak
Ya
Tampilkan pesan & Redirect ke Payment
Upload bukti pembayaran (optional)
Simpan data pesanan ke DB
Redirect ke instruksi pembayaran
Instruksi pembayaran
Upload bukti pembayaran?
Ya
Tidak
Validasi file bukti pembayaran
File valid?
Ya
Tidak
Simpan bukti pembayaran & pesan
Tampilkan pesan error (file tidak valid)
Selesai



N = 20 Simpul Predikat = 1

E = 2 Independent Path = 2 (1-2-3-4-20), (1-2-5-6-7-20),

R = 2

Kompleksitas Siklomatis = V(G) = E - N + 2

3 - 12 + 2 = 3

= V(G) = P + 1

2 + 1