

BAB 4

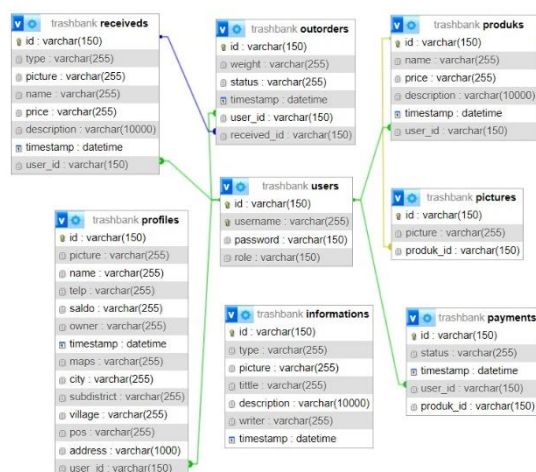
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Pengembangan aplikasi web untuk peduli sampah dan daur ulang Kota Yogyakarta merupakan aplikasi web yang dapat membantu bank sampah dalam memasarkan produk kerajinan daur ulang dan pembelian sampah daur ulang melalui platform online, membantu admin dalam memberi informasi dan edukasi kepada masyarakat, membantu admin dalam menginput data bank sampah aktif di Kota Yogyakarta, membantu admin dan bank sampah untuk menarik minat masyarakat, dan membantu masyarakat dalam memperoleh informasi serta masyarakat dapat berpartisipasi dalam program bank sampah. Rancangan penelitian yang telah disusun direalisasikan dalam bentuk implementasi pada desain antarmuka, database.

4.2 BASIS DATA

Aplikasi web ini menggunakan basis data dengan nama trashbank yang terdiri dari 8 tabel. Setiap tabel memiliki kolom "id" yang berfungsi sebagai *primary key* dan *foreign key* untuk menghubungkan antar tabel. Hubungan antar tabel ini digambarkan dalam struktur database trashbank pada Gambar 4.1.



Gambar 4.1 Implementasi Database

4.2.1 Informations

Pada Gambar 4.2 menunjukkan implementasi dari tabel *Informations*, tabel ini memiliki *id* sebagai *primary keys*.

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
☐	1	id	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐	2	type	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	3	picture	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	4	tittle	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	5	description	varchar(10000)	latin1_swedish_ci		Ya	NULL		
☐	6	writer	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	7	timestamp	datetime			Ya	NULL		

Gambar 4.2 Implementasi Tabel *Informations*

4.2.2 Outorders

Pada Gambar 4.3 menunjukkan implementasi dari tabel *outorders*, tabel ini memiliki *id* sebagai *primary key* serta *user_id* dan *received_id* sebagai *foreign key*.

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
☐	1	id	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐	2	weight	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	3	status	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	4	timestamp	datetime			Ya	NULL		
☐	5	user_id	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐	6	received_id	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.3 Implementasi Tabel *Outorders*

4.2.3 Payments

Pada Gambar 4.4 menunjukkan implementasi dari tabel *payments*, tabel ini memiliki *id* sebagai *primar key* serta *user_id* dan *produk_id* sebagai *foreign key*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
2	status	varchar(1000)	latin1_swedish_ci		Ya	NULL		
3	timestamp	datetime			Ya	NULL		
4	user_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
5	produk_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.4 Implementasi Tabel *Payments*

4.2.4 Pictures

Pada Gambar 4.5 menunjukkan implementasi dari tabel *pictures*, tabel ini memiliki *id* sebagai *primary key* dan *produk_id* sebagai *foreign key*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
2	picture	varchar(1000)	latin1_swedish_ci		Ya	NULL		
3	produk_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.5 Implementasi Tabel *Pictures*

4.2.5 Produk

Pada Gambar 4.6 menunjukkan implementasi dari tabel *produks*, tabel ini memiliki *id* sebagai *primary keys* dan *user_id* sebagai *foreign key*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
2	picture	varchar(1000)	latin1_swedish_ci		Ya	NULL		
3	picture1	varchar(1000)	latin1_swedish_ci		Ya	NULL		
4	picture2	varchar(1000)	latin1_swedish_ci		Ya	NULL		
5	name	varchar(1000)	latin1_swedish_ci		Ya	NULL		
6	price	varchar(1000)	latin1_swedish_ci		Ya	NULL		
7	description	varchar(10000)	latin1_swedish_ci		Ya	NULL		
8	timestamp	datetime			Ya	NULL		
9	user_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.6 Implementasi Tabel *Produk*s

4.2.6 Profile

Pada Gambar 4.7 menunjukkan implementasi dari tabel *profile*, tabel ini memiliki *id* sebagai *primary key* dan *user_id* sebagai *foreign key*.

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
☐	1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐	2	picture	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	3	name	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	4	telp	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	5	saldo	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	6	owner	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	7	timestamp	datetime			Ya	NULL		
☐	8	city	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	9	subdistrict	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	10	village	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	11	pos	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	12	street	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	13	address	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	14	user_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.7 Implementasi Tabel Profile

4.2.7 Receiveds

Pada Gambar 4.8 menunjukkan implementasi dari tabel *receiveds*, tabel ini memiliki *id* sebagai *primary key*.

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
☐	1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐	2	type	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	3	picture	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	4	name	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	5	price	varchar(1000)	latin1_swedish_ci		Ya	NULL		
☐	6	description	varchar(10000)	latin1_swedish_ci		Ya	NULL		
☐	7	timestamp	datetime			Ya	NULL		
☐	8	user_id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.8 Implementasi Tabel Receiveds

4.2.8 Users

Pada Gambar 4.9 menunjukkan implementasi dari tabel *users*, tabel ini memiliki *id* sebagai *primary key* dan *username* sebagai *foreign key*.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
☐ 1	id 🔑	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐ 2	username 🔑	varchar(255)	latin1_swedish_ci		Tidak	Tidak ada		
☐ 3	password	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		
☐ 4	role	varchar(150)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 4.9 Implementasi Tabel *Users*

4.3 IMPLEMENTASI SISTEM

Implementasi aplikasi web untuk peduli sampah dan daur ulang Kota Yogyakarta ini, menggunakan bahasa pemrograman Python dan pengolahan database MySQL. Berikut beberapa kode program dan hasil implementasi halaman yang terdapat pada aplikasi web untuk peduli sampah dan daur ulang Kota Yogyakarta.

4.3.1 Register

Halaman *register* atau buat akun merupakan halaman yang digunakan oleh pelanggan baru untuk mendaftarkan diri dan membuat akun baru untuk mendapatkan akses ke berbagai fitur dan layanan yang ditawarkan oleh aplikasi. Halaman *register* pada Gambar 4.10 digunakan oleh pelanggan, dengan menginput data diri lengkap seperti informasi nama, alamat, nomor telepon, *username* dan *password*.

Gambar 4.10 Implementasi Halaman *Register*

4.3.2 Login

Halaman *login* atau masuk merupakan halaman yang digunakan oleh pengguna yang sudah memiliki akun untuk masuk ke aplikasi. Halaman *login* pada Gambar 4.11 digunakan oleh admin, bank sampah, dan pelanggan dengan memasukkan *username* dan *password* yang telah terdaftar.

Gambar 4.11 Implementasi Halaman *Login*

4.3.3 Ubah Password

Halaman ubah password terdapat pada laman profile saya, halaman ubah password berfungsi sebagai wadah untuk mengubah kata sandi akun pengguna. Fitur ini memungkinkan pengguna untuk meningkatkan keamanan akun dengan

mengganti kata sandi lama pengguna dengan kata sandi baru yang lebih kuat dan sulit ditebak. Halaman ubah password pada Gambar 4.12 digunakan oleh admin, bank sampah, dan pelanggan.

Gambar 4.12 Implementasi Halaman Ubah Password

Implementasi kode program tampilan halaman Ubah Password seperti berikut:

```
def ubah_password():
    if request.method == 'POST':
        old_password = request.form.get('old_password')
        new_password = request.form.get('new_password')
        new_password1 = request.form.get('new_password1')
        if not check_password_hash(current_user.password,
old_password):
            flash('Password lama salah', 'danger')
            return redirect(url_for('auth.ubah_password'))
        if new_password != new_password1:
            flash('Konfirmasi password baru tidak cocok', 'danger')
            return redirect(url_for('auth.ubah_password'))
        current_user.password = generate_password_hash(new_password,
method='pbkdf2:sha256')
        db.session.commit()
        flash('Password berhasil diubah', 'success')
        return redirect(url_for('auth.ubah_password'))
    return render_template('profile/password.html',
current_user=current_user)
```

4.3.4 Profile

Halaman profile merupakan pusat informasi yang menampilkan data lengkap mengenai pengguna meliputi admin, bank sampah, dan pelanggan. Halaman ini berfungsi sebagai wadah identitas, profile dapat diedit jika terjadi perubahan. Pada Gambar 4.13 merupakan implementasi halaman profile.

The screenshot shows a user profile page with a header 'Home / Profile'. On the left, there is a circular profile picture of a building and the text 'Dinas Lingkungan Hidup Kota Yogyakarta' followed by address details: 'Kota Yogyakarta, Kecamatan Gondokusuman, Desa Demangan, Kode POS 55221 (Jl. Bimasakti, No.1)'. On the right, there is a form with the following fields and values:

Nama	Dinas Lingkungan Hidup Kota Yogyakarta
Username	trashbank
Nomer Telepon	0274515876
Kota	Yogyakarta,
Kecamatan	Gondokusuman
Desa	Demangan
Jalan	Bimasakti
Detail Rumah	Jl. Bimasakti, No.1

Below the form is a blue 'Edit' button.

Gambar 4.13 Implementasi Halaman Profile

4.3.5 Dashboard Admin

Halaman dashboard admin pada aplikasi web ini merupakan pusat kontrol yang dirancang khusus untuk administrator atau pengelola. Halaman dashboard admin pada Gambar 4.14 digunakan oleh admin yaitu Dinas Lingkungan Hidup Kota Yogyakarta.

The screenshot shows an admin dashboard for 'TRASH BANK'. On the left is a blue sidebar with a menu containing 'Dashboard', 'Informasi & Edukasi', and 'Toko'. The main content area is titled 'Dashboard' and contains two summary cards:

JUMLAH INFORMASI & EDUKASI	JUMLAH TOKO
1 Berita	0 Toko

At the top right of the dashboard, there is a user profile icon labeled 'ADMIN'.

Gambar 4.14 Implementasi Dashboard Admin

4.3.6 Tambah Data Informasi dan Edukasi Admin

Halaman Informasi dan Edukasi pada Admin merupakan fitur penambahan data berbagai informasi dan edukasi mengenai sampah. Penambahan data dibagi menjadi 2 kategori, yaitu data Informasi dan data Edukasi. Halaman informasi dan edukasi pada admin merupakan fitur penting yang dapat membantu meningkatkan kesadaran dan partisipasi masyarakat dalam upaya pengelolaan sampah yang berkelanjutan. Penambahan data informasi dan edukasi ditunjukkan pada Gambar 4.15.

Gambar 4.15 Implementasi Halaman Tambah Data Informasi dan Edukasi Admin

4.3.7 Tambah Data Bank Sampah

Admin mendaftarkan bank sampah dengan menginput data bank sampah seperti nama toko, alamat, nomor telepon, nama pemilik, foto toko serta username dan password. Halaman tambah data toko bank sampah yang dilakukan oleh admin ditunjukkan pada Gambar 4.16.

The image shows two web forms side-by-side. The top form, titled 'Tambah Data Bank Sampah', contains input fields for 'Kota' (pre-filled with 'Yogyakarta'), 'Kecamatan', 'Desa', 'Kode Pos', and 'Maps', followed by a large text area for 'Detail Rumah'. The bottom form, titled 'Tambah Data Toko', contains input fields for 'Username' and 'Password'. Both forms have 'Kembali' and 'Selanjutnya' or 'Add Toko' buttons at the bottom right.

Gambar 4.16 Implementasi Halaman Tambah Data Bank Sampah

Implementasi kode program admin tambah data bank sampah, seperti berikut:

```
def toko():
    if request.method == 'POST':
        name = request.form.get('storeName')
        owner = request.form.get('ownerName')
        city = request.form.get('city')
        subdistrict = request.form.get('subdistrict')
        village = request.form.get('village')
        pos = request.form.get('pos')
        address = request.form.get('address')
        username = request.form.get('username')
        password = request.form.get('password')
        telName = request.form.get('telName')
        maps = request.form.get('maps')
        picture = request.files.get('storeImage')
```

```

# Validasi data
if not all([name, owner, address, picture, username, password,
telpName, maps]):
    flash('Semua bidang harus diisi.', 'error')
    return redirect(url_for('admin.toko'))
if picture and images(picture.filename):
    user_dir = path.join(upload_images)
    if not path.exists(user_dir):
        os.makedirs(user_dir)
    extension = picture.filename.split(".")[-1]
    photo_filename = f'{name}_{owner}.{extension}'
    photo_filename = os.path.join(user_dir, photo_filename)
    photo_path = os.path.join(app.config['UPLOAD_FOLDER'],
photo_filename)
    picture.save(photo_filename)
# Check if username already exists
existing_user = User.query.filter_by(username=username).first()
if existing_user:
    flash('Username sudah terdaftar. Silakan gunakan username
lain.', category='error')
    return redirect(url_for('admin.toko'))
else :
    hashed_password = generate_password_hash(password,
method='pbkdf2:sha256')
    new_user = User(username=username, password=hashed_password,
role="shop")
    db.session.add(new_user)
    db.session.commit()
# Tambahkan data ke database
new_shop = Profile(name=name, maps=maps, city=city,
subdistrict=subdistrict, village=village, pos=pos, telp=telpName,
owner=owner, address=address, picture=photo_path, user_id=new_user.id)
    db.session.add(new_shop)
    db.session.commit()
    flash('Akun Toko berhasil ditambahkan!', 'success')

```

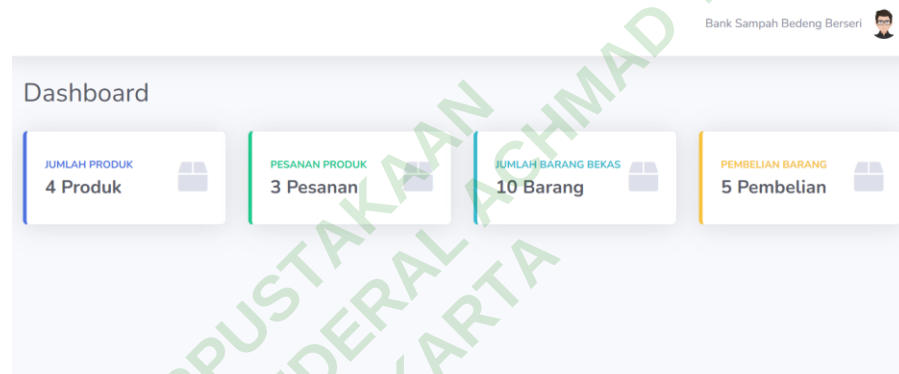
```

        return redirect(url_for('admin.toko'))
    results = db.session.query(Profile).join(User).filter(User.role ==
'shop').order_by(desc(Profile.timestamp)).all()
    return render_template('admin/toko.html', results=results)

```

4.3.8 Dashboard Bank Sampah

Halaman dashboard Bank Sampah pada Gambar 4.17 adalah halaman utama yang menampilkan informasi terkait jumlah produk, jumlah sampah, pesanan produk dan pembelian sampah. Informasi ini disajikan agar memudahkan pengelola Bank Sampah dalam mengetahui jumlah produk dan sampah yang terdata.



Gambar 4.17 Implementasi Halaman Dasboard Bank Sampah

4.3.9 List Produk Penjualan Bank Sampah

Halaman list produk penjualan bank sampah pada Gambar 4.18 adalah halaman yang menyediakan fitur tambah data produk kerajinan daur ulang untuk dipasarkan, selain menambahkan data produk, terdapat fitur edit dan hapus produk kerajinan daur ulang.

The image shows two parts of a web application. The top part is a table titled 'Produk Saya' with a '+ Tambah Data' button. The table has columns: No, Nama Produk, Harga, Deskripsi, and Aksi. It lists four products: Bunga Hias, Vas Bunga, Tas Pink, and Rompi. The bottom part is a modal form titled 'Tambah Data Produk' with fields for 'Foto Produk 1', 'Foto Produk 2', 'Nama Produk', 'Harga', and 'Deskripsi'. There are 'Choose File' buttons for the photos and an 'Hapus' button for the second photo.

No	Nama Produk	Harga	Deskripsi	Aksi
1	Bunga Hias	Rp. 35000	Bunga Hias Merr	Edit Delete
2	Vas Bunga	Rp. 25000	Vas Bunga berbe bunga palsu	Edit Delete
3	Tas Pink	Rp. 150000	Tas berbahan c formal	Edit Delete
4	Rompi	Rp. 15000	Rompi dari bahu seni. Terima requ	Edit Delete

Gambar 4.18 Implementasi Halaman Tambah Data Produk Bank Sampah

Implementasi kode program pada tampilan halaman tambah data produk kerajinan daur ulang, seperti berikut:

```
def add_produk():
    if request.method == 'POST':
        pictures = request.files.getlist('storeImage[]')
        name = request.form.get('name')
        price = request.form.get('price')
        description = request.form.get('description')
        # Validasi data
        if not all([name, price, description, pictures]):
            flash('Semua bidang harus diisi.', 'error')
            return redirect(url_for('shop.shop_produk'))
```

```

        new_produk = Produk(name=name, price=price,
description=description, user_id=current_user.id)
        db.session.add(new_produk)
        db.session.commit()
        for idx, picture in enumerate(pictures):
            if picture and images(picture.filename):
                user_dir = path.join(upload_images)
                if not path.exists(user_dir):
                    os.makedirs(user_dir)
                extension = picture.filename.split(".")[1]
                unique_id = str(uuid.uuid4())
                photo_filename =
f'{name}_{unique_id}_{price}.{extension}'
                photo_filename = os.path.join(user_dir,
photo_filename)
                picture.save(photo_filename)
                photo_path =
os.path.join(app.config['UPLOAD_FOLDER'], photo_filename)
                new_picture = Picture(picture=photo_path,
produk_id=new_produk.id)
                db.session.add(new_picture)
                db.session.commit()
                flash('Produk berhasil ditambahkan!', 'success')
                return redirect(url_for('shop.shop_produk'))

```

4.3.10 List Produk Pembelian Bank Sampah

Halaman list produk pembelian bank sampah pada Gambar 4.19 adalah halaman yang menyediakan fitur tambah data sampah yang memungkinkan pengelola bank sampah untuk menambahkan jenis sampah yang diterima oleh Bank Sampah. Fitur ini penting untuk memastikan bahwa bank sampah dapat menerima berbagai jenis sampah yang dihasilkan oleh masyarakat. Selain tambah data sampah, bank sampah juga dapat mengedit dan menghapus data sampah.

The image shows two parts of a web application. The top part is a table titled 'Daftar Barang Bekas' (List of Used Goods) with a '+ Tambah Data' button. The table has columns for No, Jenis, Nama Produk, Harga, and Aksi. It lists four items: Besi (Iron), Aluminium, Buku (Book), and Arsip (Filing Cabinet). The bottom part is a modal form titled 'Tambah Data Sampah' (Add Waste Data) with fields for Foto Sampah, Jenis Sampah, Nama Sampah, Harga, and Deskripsi, and a 'Simpan' button.

No	Jenis	Nama Produk	Harga	Aksi
1	Logam	Besi	Rp. 3500	Edit Delete
2	Logam	Aluminium	Rp. 10000	Edit Delete
3	Kertas	Buku	Rp. 1300	Edit Delete
4	Kertas	Arsip	Rp. 1800	Edit Delete

Tambah Data Sampah

Foto Sampah: No file chosen

Jenis Sampah:

Nama Sampah:

Harga:

Deskripsi:

Gambar 4.19 Implementasi Halaman Pembelian Bank Sampah

Implementasi kode program tampilan halaman tambah data sampah, seperti berikut:

```
def add_produk_received():
    if request.method == 'POST':
        picture = request.files.get('storeImage')
        type = request.form.get('type')
        name = request.form.get('name')
        price = request.form.get('price')
        description = request.form.get('description')
        # Validasi data
        if not all([type, name, price, description, picture]):
            flash('Semua bidang harus diisi.', 'error')
            return redirect(url_for('shop.received_produk'))
        if picture and images(picture.filename):
```

```

user_dir = path.join(upload_images)
if not path.exists(user_dir):
    os.makedirs(user_dir)
extension = picture.filename.split(".")[1]
photo_filename = f'{type}_{name}_{price}.{extension}'
photo_filename = os.path.join(user_dir,
photo_filename)

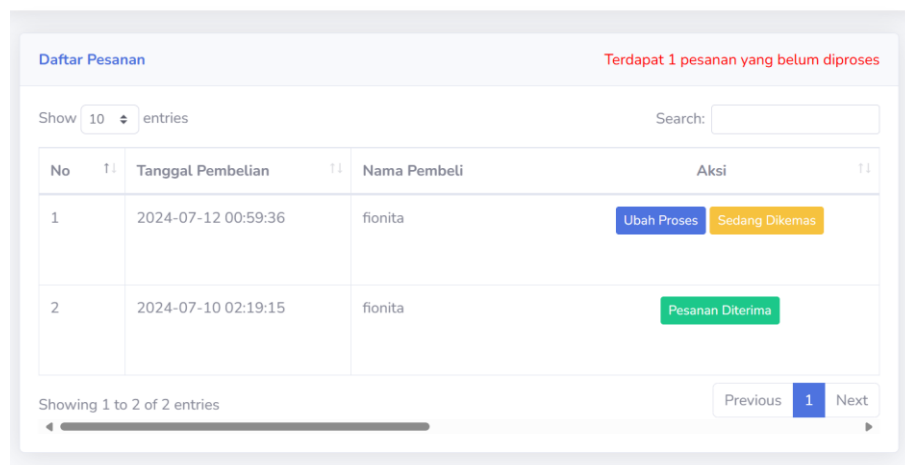
photo_path = os.path.join(app.config['UPLOAD_FOLDER'],
photo_filename)

picture.save(photo_filename)
# Tambahkan data ke database
new = Received(type=type, name=name, price=price,
description=description, picture=photo_path, user_id=current_user.id)
db.session.add(new)
db.session.commit()
flash('Produk berhasil ditambahkan!', 'success')
return redirect(url_for('shop.receive_produk'))

```

4.3.11 Pesanan Penjualan Bank Sampah

Halaman pesanan penjualan bank sampah pada Gambar 4.20 adalah fitur yang memungkinkan pengelola bank sampah untuk memantau dan mengelola semua pesanan penjualan produk kerajinan daur ulang yang telah diterima. Halaman ini menampilkan daftar pesanan lama dan baru, serta informasi detail mengenai setiap pesanan. Selain menampilkan daftar pesanan, pengelola bank sampah juga dapat mengkonfirmasi proses pesanan sesuai dengan status pesanan.

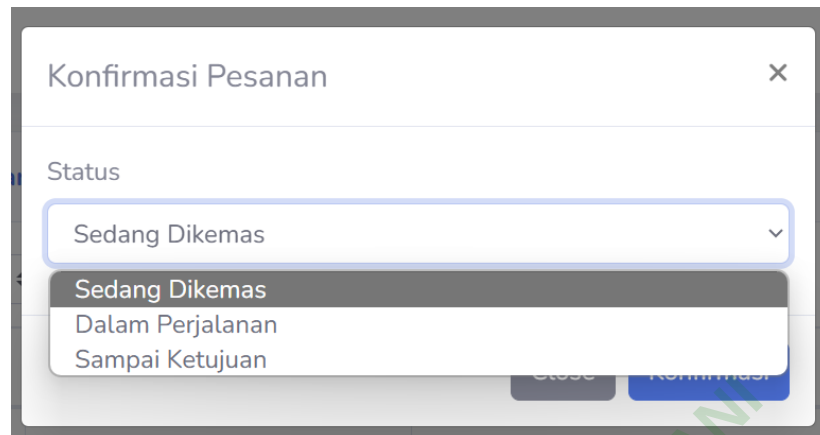


Daftar Pesanan Terdapat 1 pesanan yang belum diproses

Show 10 entries Search:

No	Tanggal Pembelian	Nama Pembeli	Aksi
1	2024-07-12 00:59:36	fionita	Ubah Proses Sedang Dikemas
2	2024-07-10 02:19:15	fionita	Pesanan Diterima

Showing 1 to 2 of 2 entries Previous 1 Next



Gambar 4.20 Implementasi Halaman Pesanan Penjualan Bank Sampah

Implementasi kode program tampilan halaman konfirmasi pesanan bank sampah, seperti berikut:

```
def konfirmasi(id):
    if request.method == 'POST':
        id = request.form.get('id')
        status = request.form.get('status')
        updated = Payment.query.filter_by(id=id).first()
        updated.status = status
        db.session.commit()
        flash('Pesanan berhasil di konfirmasi.', 'success')
        return redirect(url_for('shop.pesanan'))
```

4.3.12 Pembelian Bank Sampah

Halaman pesanan pembelian bank sampah pada Gambar 4.21 merupakan fitur yang dirancang untuk membantu pengelola bank sampah dalam mengelola transaksi pembelian sampah dari pelanggan. Halaman ini menampilkan daftar semua pesanan pembelian sampah, baik yang baru diterima maupun yang sudah lama. Pengelola dapat mengubah status pesanan sesuai dengan kondisi aktual.

Bank Sampah Berseri

Daftar Pesanan Terdapat 1 pesanan masuk yang belum dikonfirmasi

Show 10 entries Search:

No	Tanggal Penjualan	Nama Penjual	Nomer Telepon	Aksi
1	2024-07-12 01:04:27	fionita	083131313131	Ubah Proses
2	2024-07-10 02:13:14	fionita	083131313131	Transaksi Berhasil

Showing 1 to 2 of 2 entries Previous 1 Next

Konfirmasi Pesanan

Status

Menunggu Konfirmasi Bank Sampah

[Close](#) [Konfirmasi](#)

Konfirmasi Pesanan

Status

Pesanan Diterima

Berat (kg)

2

Total Harga

7000

[Close](#) [Konfirmasi](#)

Gambar 4.21 Implementasi Halaman Pesanan Pembelian Bank Sampah

Implementasi kode program tampilan halaman pesanan pembelian bank sampah, seperti berikut:

```
def konfirmasi_bekas(id):
    if request.method == 'POST':
        id = request.form.get('id')
        status = request.form.get('status')
        user_id = request.form.get('user_id')
```

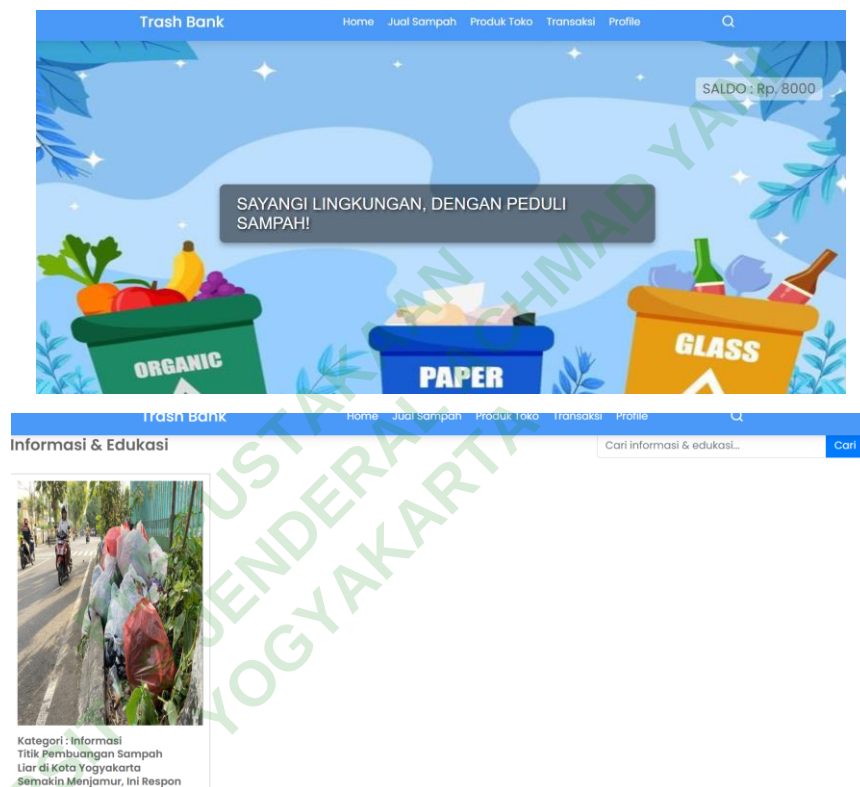
```

weight = request.form.get('weight')
display_total_price =
request.form.get('display_total_price')
if status == "Menunggu Konfirmasi Bank Sampah":
    updated = OutOrder.query.filter_by(id=id).first()
    updated.status = status
    db.session.commit()
    flash('Pesanan berhasil di konfirmasi.', 'success')
    return redirect(url_for('shop.pesanan_barang'))
elif status == "Bank Sampah Menuju Lokasi Pengambilan
Sampah":
    updated = OutOrder.query.filter_by(id=id).first()
    updated.status = status
    db.session.commit()
    flash('Pesanan berhasil di konfirmasi.', 'success')
    return redirect(url_for('shop.pesanan_barang'))
elif status == "Pesanan Diterima":
    updated = OutOrder.query.filter_by(id=id).first()
    updated.status = status
    updated.weight = weight
    update_saldo =
Profile.query.filter_by(user_id=user_id).first()
    saldo_old = float(update_saldo.saldo)
    saldo_new = float(display_total_price) + saldo_old
    update_saldo.saldo = saldo_new
    db.session.commit()
    flash('Pesanan berhasil di konfirmasi.', 'success')
    return redirect(url_for('shop.pesanan_barang'))

```

4.3.13 Home Pelanggan

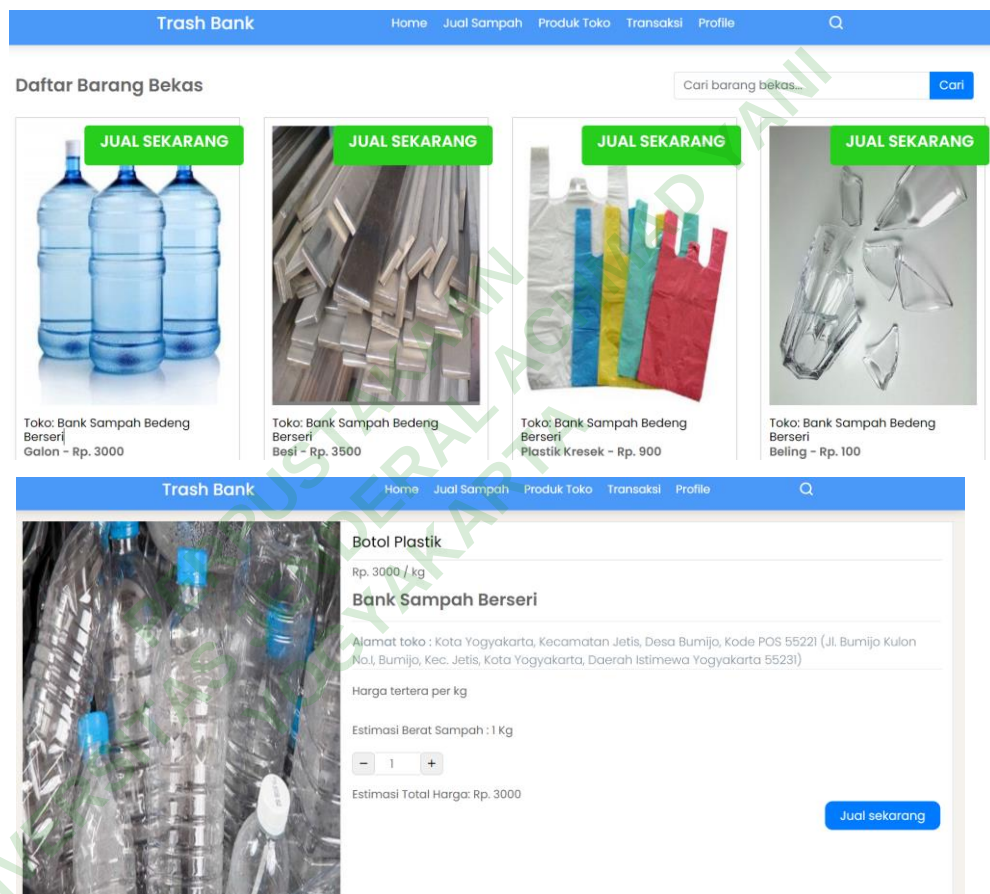
Halaman home pelanggan pada Gambar 4.22 merupakan halaman utama yang dirancang khusus untuk pelanggan bank sampah. Halaman ini berfungsi sebagai pusat informasi, edukasi, dan menampilkan berbagai fitur penting yang membantu nasabah dalam bertransaksi dan berpartisipasi dalam kegiatan bank sampah.



Gambar 4.22 Implementasi Halaman Home Pelanggan

4.3.14 Jual Sampah Pelanggan

Halaman jual sampah pelanggan pada Gambar 4.23 merupakan fitur yang dirancang untuk memudahkan pelanggan bank sampah dalam melakukan transaksi penjualan sampah. Halaman ini menyediakan informasi jenis sampah yang dapat dijual, harga per-kilogram, serta fitur-fitur yang membantu pelanggan dalam menghitung dan menyelesaikan transaksi penjualan sampah.



Gambar 4.23 Implementasi Halaman Jual Sampah Pelanggan

Implementasi kode program tampilan halaman jual sampah pelanggan, seperti berikut:

```
def orders():
    if request.method == 'POST':
        id = request.args.get('id')
        weight = request.form.get('weight')
        new_payment = OutOrder(status='Menunggu Konfirmasi Bank
Sampah', weight=weight, user_id=current_user.id, received_id=id)
```

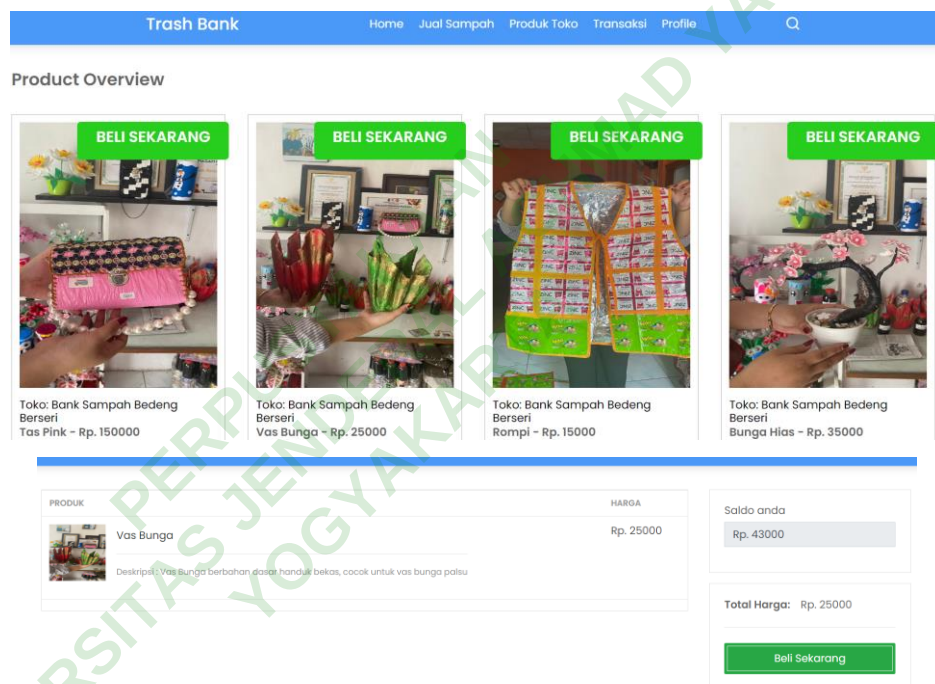
```

db.session.add(new_payment)
db.session.commit()
return redirect(url_for('home.pesanan_dijual'))

```

4.3.15 Produk Toko Pelanggan

Halaman produk toko pada Gambar 4.24 merupakan *etalase virtual* yang menampilkan berbagai produk kerajinan daur ulang dari berbagai bank sampah di Kota Yogyakarta sekaligus memberikan kesempatan bagi masyarakat untuk membeli produk ramah lingkungan yang unik dan bermanfaat.



Gambar 4.24 Implementasi Halaman Produk Toko Pelanggan

4.3.16 Detail Produk Pelanggan

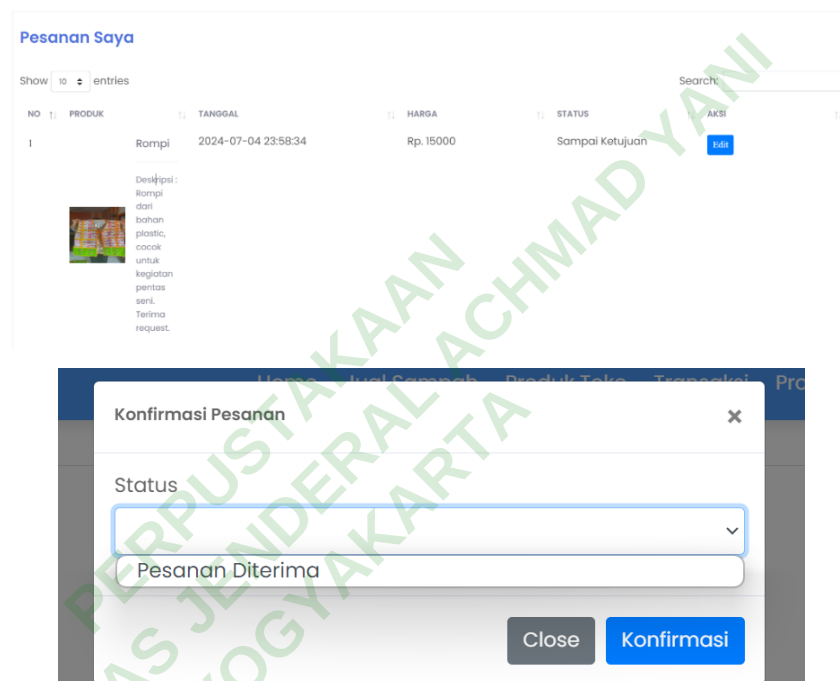
Fitur detail produk pada halaman produk toko merupakan informasi yang memberi penjelasan lengkap tentang produk kerajinan daur ulang, memungkinkan pembeli untuk menjelajahi produk dengan lebih detail sebelum membelinya. Fitur detail produk pada Gambar 4.25 memberikan gambaran yang lebih jelas tentang kualitas, keunikan, dan nilai produk.



Gambar 4.25 Implementasi Halaman Detail Produk Pelanggan

4.3.17 Transaksi Pembelian Pelanggan

Halaman transaksi pembelian saya pada Gambar 4.26 merupakan fitur yang memungkinkan pelanggan untuk mengetahui status pembelian produk kerajinan daur ulang. Fitur ini memberikan gambaran tentang aktivitas belanja, termasuk produk yang dibeli, tanggal pembelian, status pesanan, harga serta menyediakan fitur aksi untuk mengubah status pesanan jika sudah diterima.



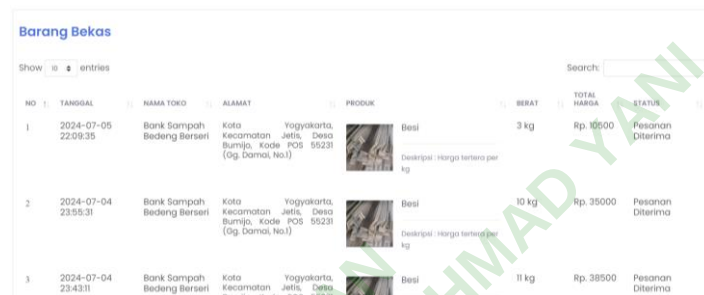
Gambar 4.26 Implementasi Halaman Transaksi Pembelian Pelanggan

Implementasi kode program tampilan halaman konfirmasi pesanan, seperti berikut:

```
def konfirmasi(id):
    if request.method == 'POST':
        id = request.form.get('id')
        status = request.form.get('status')
        updated = Payment.query.filter_by(id=id).first()
        updated.status = status
        db.session.commit()
        flash('Pesanan berhasil di konfirmasi.', 'success')
        return redirect(url_for('home.pesanan'))
```

4.3.18 Transaksi Penjualan Pelanggan

Halaman transaksi penjualan saya pada Gambar 4.27 merupakan fitur yang memungkinkan pelanggan untuk mengetahui dan mengelola riwayat penjualan sampah pelanggan kepada Bank Sampah. Fitur ini memberikan gambaran tentang aktivitas penjualan sampah, termasuk jenis sampah yang dijual, tanggal penjualan, status pesanan, alamat, nama toko, berat, dan total harga.



NO	TANGGAL	NAMA TOKO	ALAMAT	PRODUK	BERAT	TOTAL HARGA	STATUS
1	2024-07-05 22:09:35	Bank Sampah Bedeng Berseri	Kota Kecamatan Jetis, Desa Bumjo, Kode POS 55231 (lg, Damai, No.)	Besi	3 kg	Rp. 10500	Pesanan Diterima
2	2024-07-04 23:55:31	Bank Sampah Bedeng Berseri	Kota Kecamatan Jetis, Desa Bumjo, Kode POS 55231 (lg, Damai, No.)	Besi	10 kg	Rp. 35000	Pesanan Diterima
3	2024-07-04 23:43:11	Bank Sampah Bedeng Berseri	Kota Kecamatan Jetis, Desa Bumjo, Kode POS 55231 (lg, Damai, No.)	Besi	11 kg	Rp. 38500	Pesanan Diterima

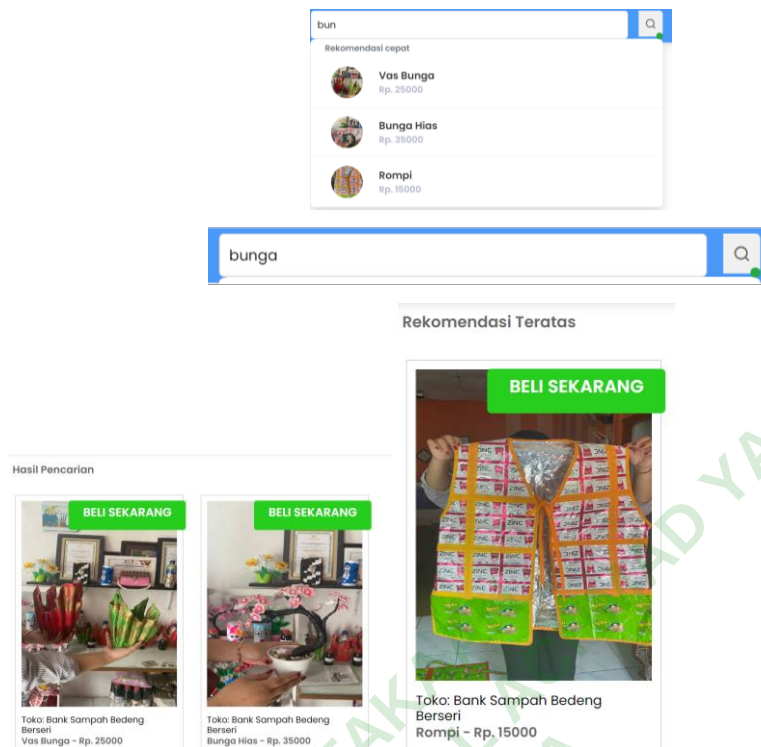
Gambar 4.27 Implementasi Halaman Transaksi Penjualan Pelanggan

4.4 CONTENT-BASED FILTERING (CBF)

CBF adalah salah satu metode dalam sistem rekomendasi untuk memberikan saran item kepada pengguna. CBF bekerja dengan cara memahami preferensi pengguna melalui item yang pernah berinteraksi sebelumnya, kemudian merekomendasikan item lain yang serupa.

4.4.1 Implementasi Content-Based Filtering (CBF)

CBF diterapkan pada halaman pencarian produk pelanggan pada Gambar 4.28 yang merupakan fitur pelanggan untuk menemukan produk kerajinan daur ulang yang diinginkan dengan mudah dan cepat serta memberikan rekomendasi produk berdasarkan minat dan kebutuhan pembeli.



Gambar 4.28 Implementasi *Content Based Filtering*

Implementasi kode program sistem rekomendasi dengan metode CBF, seperti berikut:

```
def content_based_search_with_fallback(query, produk, produk_tfidf,
vectorizer, top_n=16):
    query = preprocess_text(query)
    query_vec = vectorizer.transform([query])
    cosine_sim = cosine_similarity(query_vec, produk_tfidf)
    # Sort by cosine similarity
    sorted_indices = cosine_sim.argsort()[0][::-1]
    # Substring matching to prioritize exact matches
    substring_matches = [idx for idx, data in enumerate(produk) if
query in preprocess_text(data['name'])]
    # Remove substring matches from sorted_indices to avoid
duplication
    filtered_indices = [idx for idx in sorted_indices if idx not in
substring_matches]
    # Combine substring matches with filtered indices, prioritizing
substring matches
```

```

final_indices = substring_matches + filtered_indices[:top_n -
len(substring_matches)]
return final_indices

```

4.5 PREPROCESSING

Preprocessing adalah tahapan dalam sistem rekomendasi berbasis CBF yang bertujuan untuk membersihkan dan mempersiapkan data agar siap digunakan dalam proses selanjutnya. Berikut adalah tahapan-tahapan dalam *preprocessing*:

1. *Casefolding*

Casefolding digunakan untuk mengubah semua huruf menjadi huruf kecil untuk konsistensi agar proses selanjutnya lebih mudah. Pada Gambar 4.29 Ditunjukkan gambar hasil proses casefolding.

tempat pensil satuan = HURUF KECIL

Gambar 4.29 Hasil Proses *Casefolding*

2. *Tokenizing*

Tokenizing atau tokenisasi bertujuan untuk memisahkan teks kalimat menjadi unit-unit individu (token) agar kat-kata tersebut dapat berdiri sendiri. Pada Gambar 4.30 Ditunjukkan gambar hasil proses *tokenizing*.

tempat pensil satuan = MEMISAHKAN KATA

Gambar 4.30 Hasil Proses *Tokenizing*

3. *Cleaning*

Cleaning bertujuan untuk membersihkan data dari elemen yang tidak relevan atau tidak bermakna, seperti tanda baca, simbol, angka, dan karakter numeik lainnya. Pada Gambar 4.31 Ditunjukkan gambar hasil proses *cleaning*.

tempat pensil satuan = MENGHILANGKAN TANDA BACA, SIMBOL DAN ANGKA

Gambar 4.31 Hasil Proses *Cleaning*

4. *Stopwords*

Stopwords bertujuan untuk menghapus kata-kata umum yang tidak memiliki makna signifikan dari teks, seperti "dan", "yang", "ini", dll. Pada Gambar 4.32 Ditunjukkan gambar hasil proses *stopwords*.

pensil = MERUBAH KATA YANG SERING DIPAKAI TAPI TIDAK BERARTI BANYAK

Gambar 4.32 Hasil Proses *Stopwords*

5. *Stemming*

Stemming bertujuan untuk mengubah kata menjadi bentuk dasarnya (akar kata). Pada Gambar 4.33 Ditunjukkan gambar hasil proses *stemming*.

tempat pensil satu = MENGUBAH KATA MENJADI KATA DASAR

Gambar 4.33 Hasil Proses *Stemming*

4.6 TERM FREQUENCY-INVERS DOCUMENT FREQUENCY (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan sebuah metrik yang digunakan untuk mengukur keberartian suatu kata terhadap suatu dokumen dalam sebuah kumpulan dokumen. Tabel 4.1 menunjukkan keterangan dokumen untuk pembobotan TF-IDF.

Tabel 4.1 Keterangan Dokument TF-IDF

Document (d)	Keterangan
d1	bunga
d2	sandal
d3	vas
d4	dompet
d5	rompi
d6	tabungan
d7	tas
d8	kursi
d9	meja
d10	tikar

Tabel 4.2 menunjukkan hasil perhitungan *term frequency* (TF), untuk setiap kata kunci dalam kumpulan dokumen yang dianalisis dalam penelitian ini. perhitungan TF penting untuk melanjutkan ke tahap berikutnya.

Tabel 4.2 Perhitungan *Term Frequency* (TF)

<i>term/kata</i>	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	df(t)
bunga	1										2
dompet				1							2
kursi								1			2
meja									1		2
rompi					1						2
sandal		1									2
tabungan						1					1
tas							1				2
tikar										1	2
vas			1								2

Tahap selanjutnya adalah perhitungan *inverse document frequency* (IDF) yang ditunjukkan pada Tabel 4.3. *Output* IDF yang dihasilkan adalah sebagai berikut.

Tabel 4.3 Perhitungan *Inverse Document Frequency* (IDF)

<i>term/kata</i>	df(t)	idf
bunga	2	1,1513
dompet	2	1,1513
kursi	2	1,1513
meja	2	1,1513
rompi	2	1,1513
sandal	2	1,1513
tabungan	1	1,1513
tas	2	1,1513
tikar	2	1,1513
vas	2	1,1513

Tabel 4.4 menunjukkan hasil perhitungan *term frequency-inverse document frequency* (TF-IDF) nilai TF-IDF yang mendekati 1 menunjukkan bahwa suatu kata memiliki keberartian yang sangat tinggi terhadap suatu dokumen. Hal ini dapat mengindikasikan bahwa kata tersebut merupakan topik utama, unik, dan memiliki informasi yang penting untuk dokumen tersebut. Nilai TF-IDF dihitung dengan mengalikan frekuensi istilah (TF) dengan invers frekuensi dokumen (IDF).

Tabel 4.4 Perhitungan *Term Frequency-Inverse Document Frequency* (TF-IDF)

term/kata	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	df(t)
bunga	1,00										2
dompet				1,00							2
kursi								1,00			2
meja									1,00		2
rompi					1,00						2
sandal		1,00									2
tabungan						1,00					1
tas							1,00				2
tikar										1,00	2
vas			1,00								2

4.7 PENGUJIAN SISTEM

Sebelum diluncurkan atau digunakan secara resmi maka diperlukan pengujian sistem pada aplikasi web untuk peduli sampah dan daur ulang Kota Yogyakarta guna mengetahui apakah aplikasi web tersebut memenuhi kebutuhan dan tujuan yang telah ditetapkan. Pengujian sistem meliputi *confusion matrix*, *blackbox*, dan UAT.

4.7.1 Confusion Matrix

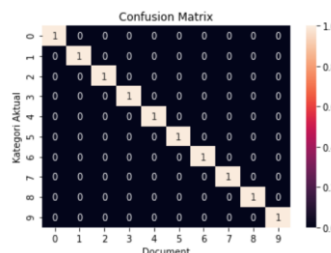
Confusion Matrix akan digunakan untuk menghitung *support*, *precision*, *recall*, *f1_score*, dan *accuracy*. Perhitungan *confusion matrix* akan memberikan gambaran yang komprehensif mengenai kemampuan sistem dalam mengklasifikasikan data dengan benar.

4.7.1.1 Visualisasi Confusion Matrix

Kinerja sistem rekomendasi CBF yang digunakan dalam aplikasi web peduli sampah akan dievaluasi secara menyeluruh dengan menerapkan pengujian menggunakan *confusion matrix* terhadap data yang digunakan. Melalui pengujian *confusion matrix*, dapat memperoleh gambaran yang komprehensif dalam mengklasifikasikan data dengan benar. Berikut kode untuk visualisasi *confusion matrix* :

```
sns.heatmap(matriks_konfusio, annot=True, fmt="d")
plt.xlabel("Document")
plt.ylabel("Kategori Aktual")
plt.title("Confusion Matrix")
plt.show()
```

Pada Gambar 4.34 akan menunjukkan hasil *heatmap* visualisasi *confusion matrix*, untuk klasifikasi dokumen. *Confusion matrix* ini membantu dalam mengevaluasi performa model klasifikasi dalam memprediksi kategori dokumen. Sumbu X: Menunjukkan kategori prediksi model (dalam hal ini, kategori prediksi identik dengan ID dokumen). Sumbu Y: Menunjukkan kategori aktual dokumen (misalnya, "bunga", "sandal", "vas", dan seterusnya). Nilai pada kotak: Menunjukkan jumlah dokumen yang diklasifikasikan ke dalam kategori tertentu. Warna yang lebih gelap menunjukkan jumlah yang lebih banyak.



Gambar 4.34 Heatmap Confusion Matrix

4.7.1.2 Perhitungan Matrix Confusion

Selanjutnya adalah menghitung *Confusion matrix*, dengan menggunakan kode program berikut:

```
def evaluate_recommendations(recommended, relevant):
    y_true = [1 if item in relevant else 0 for item in recommended]
    y_pred = [1 if item in relevant else 0 for item in recommended]
    controlled_noise_indices = random.sample(range(len(y_pred)),
max(1, int(0.1 * len(y_pred))))
    for idx in controlled_noise_indices:
        y_pred[idx] = 1 - y_pred[idx]
    precision = precision_score(y_true, y_pred, zero_division=0) * 100
    recall = recall_score(y_true, y_pred, zero_division=0) * 100
    f1 = f1_score(y_true, y_pred, zero_division=0) * 100
    support = sum(y_true) # number of relevant items
    accuracy = accuracy_score(y_true, y_pred) * 100
    print(f"Support: {support}, Presisi: {precision:.2f}%, Recall:
{recall:.2f}%, F1-Score: {f1:.2f}%, Akurasi: {accuracy:.2f}%")
    if y_true and y_pred:
        print(classification_report(y_true, y_pred,
target_names=['not relevant', 'relevant'], labels=[0, 1],
zero_division=0))

    metrics = {
        'Precision': precision,
        'Recall': recall,
        'F1-Score': f1,
        'Accuracy': accuracy
    }
    metric_names = list(metrics.keys())
    metric_values = list(metrics.values())

    plt.figure(figsize=(10, 6))
    plt.bar(metric_names, metric_values, color=['blue', 'green',
'red', 'purple'])
    plt.xlabel('Metrics')
    plt.ylabel('Percentage')
```

```

plt.title('Evaluation Metrics')
plt.ylim(0, 100)
for i, v in enumerate(metric_values):
    plt.text(i, v + 1, f"{v:.2f}%", ha='center', va='bottom')
plot_path = os.path.join(upload_images, 'metrics_plot.png')
plt.savefig(plot_path)
plt.close()
print(f"Plot saved at: {plot_path}")
return metrics, plot_path

```

Support dihitung dengan menjumlahkan item relevan (y_{true}) untuk digunakan menghitung *accuracy*. *Precision* dihitung menggunakan *precision_score* dari library *scikit-learn* untuk mengukur *percent* rekomendasi yang relevan dari total rekomendasi yang diberikan. *Recall* dihitung menggunakan *recall_score* dari library *scikit-learn* untuk mengukur *percent* item relevan yang berhasil direkomendasikan. *F1* dihitung menggunakan *F1_Score* dari library *scikit-learn* untuk menggabungkan *precision* dan *recall* untuk menilai *accuracy* keseluruhan. *Accuracy* dihitung menggunakan *accuracy_score* dari library *scikit-learn* untuk mengukur *percent* prediksi yang benar (relevan atau tidak relevan) dibandingkan dengan total item. Pada Tabel 4.5 adalah hasil dari pengujian sistem CBF dengan menggunakan metode *confusion matrix*.

Tabel 4.5 Hasil Perhitungan *Confusion Matrix*

Matrix	Hasil
<i>Support</i>	3
<i>Precision</i>	75,00%
<i>Recall</i>	100,00%
<i>F1_Score</i>	85,71%
<i>Accuracy</i>	93,33%

Berdasarkan hasil perhitungan *confusion matrix* pada Tabel 4.6, dapat disimpulkan bahwa sistem rekomendasi dengan metode CBF pada aplikasi web Peduli Sampah dan Daur Ulang Kota Yogyakarta memiliki performa yang baik. Aplikasi ini mampu merekomendasikan item yang relevan dengan kebutuhan pengguna dengan *precision* dan *recall* yang tinggi. *Accuracy* aplikasi tergolong

tinggi, menunjukkan bahwa aplikasi ini dapat memprediksi dengan benar apakah suatu item relevan dengan pengguna atau tidak.

4.7.2 *Blackbox Testing*

Blackbox testing dilakukan dengan 2 pakar, dengan menguji fungsionalitas fitur-fitur aplikasi web peduli sampah dan daur ulang Kota Yogyakarta. Terdapat 35 fitur yang diujikan, pada Tabel 4.6 ditunjukkan hasil perhitungan dari pengujian sistem dengan metode *blackbox testing*.

Tabel 4.6 *Blackbox Testing*

Aktor	Test Case	Hasil (%)
Admin	11	100%
Bank Sampah	15	100%
Pelanggan	12	100%

Tabel 4.6 menunjukkan hasil 100% pada pengujian sistem dengan metode *blackbox*, sehingga dapat disimpulkan bahwa fitur-fitur yang terdapat pada aplikasi web peduli sampah dan daur ulang Kota Yogyakarta dapat berjalan dengan baik sesuai dengan fungsionalitas yang diharapkan.

4.7.3 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah tahap pengujian pengembangan aplikasi, dimana pengguna (*end-user*) mengevaluasi aplikasi web ini. Tujuan UAT dilakukan adalah untuk memastikan aplikasi web memenuhi kebutuhan dan harapan para pengguna sebelum dirilis atau digunakan secara resmi, pengujian UAT menggunakan kuesioner dengan isi pernyataan menggunakan metode *USE Questionnaire: Usefulness, Satisfaction, and Ease of use*. Kuesioner *USE Questionnaire* ini dirancang untuk mengukur tingkat kegunaan, kepuasan, dan kemudahan penggunaan aplikasi web oleh pengguna. Pengujian UAT dilakukan kepada 10 masyarakat, 3 bank sampah dan 2 admin, melalui kuesioner dengan skala penilaian, maka dapat diketahui apakah aplikasi web ini layak digunakan dan sesuai kebutuhan pengguna. Pada Tabel 4.7 ditunjukkan keterangan penilaian *skala likert* (Purba et al., 2021).

Tabel 4.7 Keterangan Penilaian *Skala Likert*

<i>Skala Likert</i>	Keterangan	Nilai
1	Tidak Setuju	0 – 25 %
2	Kurang Setuju	26 – 50 %
3	Setuju	51 – 75 %
4	Sangat Setuju	76 – 100 %

Pengujian UAT dengan metode *USE Questionnaire* menilai empat aspek utama kegunaan aplikasi, yaitu: *Usefulness* atau kegunaan, *Ease of Use* atau kemudahan penggunaan, *Ease of Learn* atau kemudahan dipelajari, dan *Satisfaction* kepuasan. Pengujian UAT dilakukan dengan menyebar kuesioner ke 15 responden, meliputi: 2 Admin, 3 Bank Sampah, dan 10 Masyarakat, sehingga menghasilkan nilai yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian UAT

Aktor	Faktor	Pernyataan	1	2	3	4
Admin	U	Aplikasi ini membantu saya dalam mendaftarkan Bank Sampah			1	1
	U	Aplikasi ini membantu saya dalam membagikan Informasi dan Edukasi kepada masyarakat			1	1
	Total <i>Usefulness</i>				2	2
	EoU	Saya merasa nyaman menggunakan aplikasi ini dan tidak mengalami kesulitan yang berarti.				2
	EoU	Aplikasi ini mudah dimengerti dan user friendly				2
	Total <i>Ease of Use</i>					4
	EoL	Saya dapat dengan mudah mempelajari cara menggunakan aplikasi ini			1	1
	EoL	Saya tidak membutuhkan banyak waktu untuk			2	

		mempelajari cara menggunakan aplikasi ini				
	EoL	Saya merasa aplikasi ini mudah dipelajari bahkan bagi orang yang tidak terbiasa dengan teknologi			2	
	<i>Total Ease of Learn</i>				5	1
	S	Saya merasa puas dengan kemudahan penggunaan aplikasi ini.			1	1
	S	Saya merasa puas dengan layanan yang ditawarkan oleh aplikasi ini			1	1
	S	Saya merasa puas dengan desain dan tampilan aplikasi ini			2	
	<i>Total Satisfaction</i>				4	2
Bank Sampah	U	Aplikasi ini membantu saya dalam menjual produk kerajinan daur ulang			1	2
	U	Aplikasi ini membantu saya dalam menerima pesanan pembelian sampah			1	2
	U	Aplikasi ini membantu saya dalam memproses pesanan pelanggan			1	2
	<i>Total Usefulness</i>				3	6
	EoU	Saya merasa nyaman menggunakan aplikasi ini dan tidak mengalami kesulitan yang berarti.			1	2
	EoU	Aplikasi ini mudah dimengerti dan user friendly.			1	2
	<i>Total Ease of Use</i>				2	4
	EoL	Saya dapat dengan mudah mempelajari cara menggunakan aplikasi ini			3	

	EoL	Saya tidak membutuhkan banyak waktu untuk mempelajari cara menggunakan aplikasi ini			3	
	EoL	Saya merasa aplikasi ini mudah dipelajari bahkan bagi orang yang tidak terbiasa dengan teknologi			2	1
	Total Ease of Learn				8	1
	S	Saya merasa puas dengan kemudahan penggunaan aplikasi ini.			1	2
	S	Saya merasa puas dengan layanan yang ditawarkan oleh aplikasi ini			1	2
	S	Saya merasa puas dengan desain dan tampilan aplikasi ini			1	2
	Total Satisfaction				3	6
Pelanggan	U	Aplikasi ini menyediakan informasi dan edukasi yang berguna mengenai sampah			7	3
	U	Aplikasi ini membantu saya dalam menjual sampah daur ulang			5	5
	U	Aplikasi ini membantu saya dalam membeli produk kerajinan daur ulang			7	3
	Total Usefulness				19	11
	EoU	Saya merasa nyaman menggunakan aplikasi ini dan tidak mengalami kesulitan yang berarti.		1	2	7
	EoU	Aplikasi ini mudah dimengerti dan user friendly.			4	6
	Total Ease Of Use			1	6	13

	EoL	Saya dapat dengan mudah mempelajari cara menggunakan aplikasi ini		1	5	4
	EoL	Saya tidak membutuhkan banyak waktu untuk mempelajari cara menggunakan aplikasi ini			3	7
	EoL	Saya merasa aplikasi ini mudah dipelajari bahkan bagi orang yang tidak terbiasa dengan teknologi		1	6	3
	Total Ease of Learn			2	14	14
	S	Saya merasa puas dengan kemudahan penggunaan aplikasi ini.			5	5
	S	Saya merasa puas dengan layanan yang ditawarkan oleh aplikasi ini			7	3
	S	Saya merasa puas dengan desain dan tampilan aplikasi ini			8	2
	S	Saya merasa puas dengan informasi yang disediakan oleh aplikasi ini			4	6
	Total Satisfaction				24	16

Setelah mengetahui hasil penilaian UAT, maka akan dilakukan perhitungan atau pengukuran UAT dengan tujuan untuk menganalisis data kuesioner, dengan menggunakan rumus berikut (Ripai, 2023):

$$P = \frac{S}{Skor Ideal} \times 100\%$$

Keterangan :

P = Percent (%).

S = Skor setiap jawaban.

Skor ideal = Skor tertinggi dikalikan dengan jumlah responden dan jumlah pernyataan.

1. Perhitungan *USE Admin*

Berikut adalah hasil perhitungan data *usefulness*, *ease of use*, *ease of learn*, and *satisfaction* pada admin.

- Hasil perhitungan *usefulness* pada admin ditunjukkan pada Tabel 4.9

Tabel 4.9 *Usefulness Admin*

Penguji	Skor per Item Kuesioner (Skala 1-4)	
	1	2
Admin 1	3	3
Admin 2	4	4
	7	7
Total Usefulness	7+7 = 14	

$$P = \frac{14}{16} \times 100\%$$

$$= 87,5\%$$

- Hasil perhitungan *ease of use* pada admin, ditunjukkan pada Tabel 4.10

Tabel 4.10 *Ease of Use Admin*

Penguji	Skor per Item Kuesioner (Skala 1-4)	
	1	2
Admin 1	4	4
Admin 2	4	4
	8	8
Total Ease of Use	8 + 8 = 16	

$$P = \frac{16}{16} \times 100\%$$

$$= 100\%$$

- Hasil perhitungan *ease of learn* pada admin, ditunjukkan pada Tabel 4.11

Tabel 4.11 *Ease of Learn Admin*

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Admin 1	3	3	3
Admin 2	4	3	3
	7	6	6
Total Ease of Use	7 + 6 + 6 = 19		

$$P = \frac{19}{24} \times 100\%$$

$$= 79,1\%$$

- Hasil perhitungan *satisfaction* pada admin, ditunjukkan pada Tabel 4.12

Tabel 4.12 *Satisfaction Admin*

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Admin 1	3	3	3
Admin 2	4	4	3
	7	7	6
Total Satisfaction	7 + 7 + 6 = 19		

$$P = \frac{19}{24} \times 100\%$$

$$= 79,1\%$$

- Hasil perhitungan *usefulness*, *ease of use*, *ease of learn*, dan *satisfaction* pada admin ditunjukkan pada tabel 4.13.

Tabel 4.13 Perhitungan USE Admin

Penguji	Skor per Item Kuesioner (Skala 1-4)									
	1	2	3	4	5	6	7	8	9	10
Admin 1	3	3	4	4	3	3	3	4	3	3
Admin 2	4	4	4	4	4	3	3	3	4	3
	7	7	8	8	7	6	6	7	7	6
Total USE	7 + 7 + 8 + 8 + 7 + 6 + 6 + 7 + 7 + 6 = 69									

$$P = \frac{69}{80} \times 100\%$$

$$= 86,2\%$$

2. Perhitungan USE Bank Sampah

- Hasil perhitungan *usefulness* pada bank sampah ditunjukkan pada Tabel 4.14

Tabel 4.14 *Usefulness* Bank Sampah

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Bank Sampah 1	3	3	3
Bank Sampah 2	4	4	4
Bank Sampah 3	4	4	4
	11	11	11
Total Usefulness	11 + 11 + 11 = 33		

$$P = \frac{33}{36} \times 100\%$$

$$= 91,6\%$$

- Hasil perhitungan *ease of use* pada bank sampah ditunjukkan pada Tabel 4.15

Tabel 4.15 *Ease of Use* Bank Sampah

Penguji	Skor per Item Kuesioner (Skala 1-4)	
	1	2
Bank Sampah 1	3	3
Bank Sampah 2	4	4
Bank Sampah 3	4	4
	11	11
Total Ease of Use	11 + 11 = 22	

$$P = \frac{22}{24} \times 100\%$$

$$= 91,6\%$$

- Hasil perhitungan *ease of learn* pada bank sampah ditunjukkan pada Tabel 4.16

Tabel 4.16 *Ease of Learn* Bank Sampah

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Bank Sampah 1	3	3	3
Bank Sampah 2	3	3	3
Bank Sampah 3	3	3	4
	9	9	10
Total Ease of Learn	9 + 9 + 10 = 28		

$$P = \frac{28}{36} \times 100\%$$

$$= 77,7\%$$

- Hasil perhitungan *satisfaction* pada bank sampah ditunjukkan pada Tabel 4.17

Tabel 4.17 *Satisfaction* Bank Sampah

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Bank Sampah 1	3	4	4
Bank Sampah 2	4	3	4
Bank Sampah 3	4	4	3
	11	11	11
Total Satisfaction	11 + 11 + 11 = 33		

$$P = \frac{33}{36} \times 100\%$$

$$= 91,6\%$$

- Hasil perhitungan *usefulness*, *ease of use*, *ease of learn*, dan *satisfaction* pada bank sampah ditunjukkan pada Tabel 4.18.

Tabel 4.18 Perhitungan USE Bank Sampah

Penguji	Skor per Item Kuesioner (Skala 1-4)										
	1	2	3	4	5	6	7	8	9	10	11
Bank Sampah 1	3	4	4	3	3	3	3	3	3	4	4
Bank Sampah 2	4	3	4	4	4	3	3	3	4	3	4
Bank Sampah 3	4	4	3	4	4	3	3	4	4	4	3
	11	11	11	11	11	9	9	10	11	11	11
Total USE	11 + 11 + 11 + 11 + 11 + 9 + 9 + 10 + 11 + 11 + 11 = 116										

$$P = \frac{116}{132} \times 100\%$$

$$= 87,8\%$$

3. Perhitungan USE Pelanggan

- Hasil perhitungan *usefulness* pada pelanggan ditunjukkan pada Tabel 4.19

Tabel 4.19 *Usefulness* Pelanggan

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Pelanggan 1	3	4	3
Pelanggan 2	3	4	3
Pelanggan 3	3	4	3
Pelanggan 4	3	4	3
Pelanggan 5	3	4	3
Pelanggan 6	3	3	3
Pelanggan 7	3	3	3
Pelanggan 8	4	3	4
Pelanggan 9	4	3	4
Pelanggan 10	4	3	4
	33	35	33
Total Usefulness	33 + 35 + 33 = 101		

$$\begin{aligned}
 P &= \frac{101}{120} \times 100\% \\
 &= 84,1\%
 \end{aligned}$$

- Hasil perhitungan *ease of use* pada pelanggan ditunjukkan pada Tabel 4.20

Tabel 4.20 *Ease of Use* Pelanggan

Penguji	Skor per Item Kuesioner (Skala 1-4)	
	1	2
Pelanggan 1	2	3
Pelanggan 2	3	3
Pelanggan 3	3	3
Pelanggan 4	4	3
Pelanggan 5	4	4
Pelanggan 6	4	4
Pelanggan 7	4	4
Pelanggan 8	4	4
Pelanggan 9	4	4
Pelanggan 10	4	4
	36	36
Total <i>Ease of Use</i>	36 + 36 = 72	

$$\begin{aligned}
 P &= \frac{72}{80} \times 100\% \\
 &= 90\%
 \end{aligned}$$

- Hasil perhitungan *ease of learn* pada pelanggan ditunjukkan pada Tabel 4.21

Tabel 4.21 *Ease of Learn* Pelanggan

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Pelanggan 1	2	3	2
Pelanggan 2	3	3	3
Pelanggan 3	3	3	3
Pelanggan 4	3	4	3
Pelanggan 5	3	4	3
Pelanggan 6	3	4	3
Pelanggan 7	4	4	3
Pelanggan 8	4	4	4
Pelanggan 9	4	4	4
Pelanggan 10	4	4	4
	33	37	32
Total Ease of Learn	33 + 37 + 32 = 102		

$$\begin{aligned}
 P &= \frac{102}{120} \times 100\% \\
 &= 85\%
 \end{aligned}$$

- Hasil perhitungan *satisfaction* pada pelanggan ditunjukkan pada Tabel 4.22

Tabel 4.22 *Satisfaction* Pelanggan

Penguji	Skor per Item Kuesioner (Skala 1-4)		
	1	2	3
Pelanggan 1	3	3	4
Pelanggan 2	3	3	4
Pelanggan 3	3	3	3
Pelanggan 4	3	3	3
Pelanggan 5	3	3	3
Pelanggan 6	4	3	3
Pelanggan 7	4	3	3
Pelanggan 8	4	4	3
Pelanggan 9	4	4	3
Pelanggan 10	4	4	3
	35	33	32
Total Satisfaction	35 + 33 + 32 = 100		

$$\begin{aligned}
 P &= \frac{100}{120} \times 100\% \\
 &= 83,3\%
 \end{aligned}$$

- Hasil perhitungan *usefulness*, *ease of use*, *ease of learn*, dan *satisfaction* pada pelanggan ditunjukkan pada Tabel 4.23.

Tabel 4.23 Perhitungan USE Pelanggan

Penguji	Skor per Item Kuesioner (Skala 1-4)										
	1	2	3	4	5	6	7	8	9	10	11
Pelanggan 1	3	3	4	4	3	4	3	2	3	4	3
Pelanggan 2	3	3	4	4	3	4	3	4	3	4	3
Pelanggan 3	3	3	4	4	3	4	3	4	3	4	3
Pelanggan 4	3	3	4	4	3	4	4	4	3	3	3
Pelanggan 5	3	3	4	4	4	3	4	3	3	3	3
Pelanggan 6	3	4	4	4	4	3	4	3	4	3	3
Pelanggan 7	3	4	4	4	4	3	4	3	4	3	3
Pelanggan 8	4	4	3	3	4	3	4	3	4	3	3
Pelanggan 9	4	4	3	3	4	3	4	3	4	3	4
Pelanggan 10	4	4	3	2	4	2	4	3	4	3	4
	33	35	37	36	36	30	37	32	35	33	32
Total USE	33 + 35 + 37 + 36 + 36 + 30 + 37 + 32 + 35 + 33 + 32 = 376										

$$P = \frac{376}{440} \times 100\%$$

$$= 85,4\%$$

Berdasarkan pengujian UAT dengan metode *USE Questionnaire* pada aplikasi web peduli sampah dan daur ulang Kota Yogyakarta, dengan melibatkan 15 responden yang meliputi 3 aktor utama yaitu admin, bank sampah, dan pelanggan serta analisis hasil kuesioner menunjukkan USE admin memperoleh nilai 86,2%; USE bank sampah memperoleh nilai 87,8%; dan USE pelanggan memperoleh nilai 85,4%. Sehingga nilai rata rata perhitungan USE dari 3 aktor

memperoleh 86,6% dari 100%, berdasarkan nilai rata rata menunjukkan hasil aplikasi web peduli sampah dan daur ulang Kota Yogyakarta sangat layak untuk digunakan.

4.8 PEMBAHASAN

Aplikasi web Peduli Sampah dan Daur Ulang Kota Yogyakarta dikembangkan untuk membantu bank sampah agar dapat memasarkan produk kerajinan daur ulang secara online, memperluas jangkauan pasar dan meningkatkan pendapatan. Selain itu, aplikasi ini memberikan layanan transaksi online antara bank sampah dan pelanggan untuk pembelian sampah, sehingga memudahkan masyarakat dalam berpartisipasi dalam program daur ulang. Bagi pelanggan, aplikasi ini menyediakan informasi dan edukasi mengenai sampah dan pengelolaannya, pelanggan juga dapat berbelanja produk kerajinan daur ulang secara online, mendukung ekonomi kreatif dan berkontribusi terhadap pengurangan sampah. Aplikasi web ini juga memberikan manfaat bagi admin, dengan fitur pendaftaran bank sampah dan untuk membagikan informasi dan edukasi kepada pelanggan. Berdasarkan hasil pengujian sistem *confusion matrix*, *blackbox* dan UAT dapat disimpulkan performa aplikasi web peduli sampah dan daur ulang Kota Yogyakarta layak dan mencapai tujuan awal pengembangan.

Pengembangan aplikasi web peduli sampah dan daur ulang kota Yogyakarta menerapkan metode *prorotyping*, tahap pertama yaitu *communication* untuk menganalisis kebutuhan DLH Kota Yogyakarta sehingga dapat diketahui kebutuhan fungsional untuk aplikasi. Tahap kedua yaitu *quick planning*, setelah mengetahui kebutuhan DLH selanjutnya adalah membuat perancangan cepat untuk menentukan spesifikasi input, output, dan proses yang diperlukan seperti data bank sampah, pelanggan, dan admin. Tahap ketiga yaitu *modeling quick design*, bertujuan untuk menggambarkan alur kinerja aplikasi dan pengguna melalui perancangan diagram UML dan desain antarmuka pengguna (UI). Tahap keempat yaitu *Construction of Prototype*, mengubah desain UI kedalam kode program python dan menyimpan data kedalam database MySQL. Tahap kelima adalah *Deployment, Delivery & Feedback*, sebelum menyerahkan aplikasi kepada

pengguna maka dilakukan pengujian sistem menggunakan *confusion matrix*, *blackbox*, UAT. *Feedback* atau umpan balik dari DLH dilakukan dengan cara evaluasi aplikasi bersama, untuk mendapatkan umpan balik kembali. Hasil evaluasi bersama akan dilakukan analisis kembali sehingga proses ini akan terus berulang sampai mencapai keinginan pengguna, proses berulang disebut juga *iterasi prototyping*.

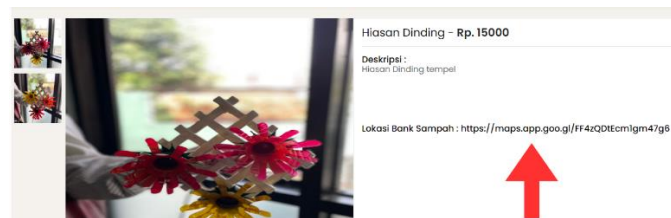
4.8.1 Iterasi Prototyping

Iterasi prototyping pada Aplikasi web peduli sampah dan daur ulang Kota Yogyakarta yang dilatarbelakangi oleh masukan Dinas Lingkungan Hidup Kota Yogyakarta melalui tahap *communication* yaitu penambahan fitur lokasi pada bank sampah. Masukan tersebut menjadi dasar untuk melakukan *iterasi prototyping* dan menambahkan fitur lokasi bank sampah. Fitur ini memungkinkan pengguna untuk mengetahui dan menemukan lokasi bank sampah, pada Gambar 4.35 menunjukkan penambahan fitur lokasi bank sampah.

1. Sebelum Iterasi Prototyping



2. Setelah Iterasi Prototyping



Gambar 4.35 Implementasi *Iterasi Prototyping*

Pada Gambar 4.35 terjadi *iterasi prototyping* pada halaman detail produk, Fitur ini dikembangkan berdasarkan masukan dari pengguna selama tahap pengujian awal. Pengguna menyatakan bahwa perlunya mengembangkan fitur lokasi untuk pelanggan dapat menemukan informasi tentang lokasi bank sampah, yang menjadi salah satu kekurangan dalam menggunakan aplikasi. *Iterasi prototyping* diwujudkan secara nyata melalui penambahan fitur lokasi bank sampah, fitur ini menjawab kebutuhan dan keinginan Dinas Lingkungan Hidup Kota Yogyakarta secara langsung dan efisien. Pada penelitian ini, *iterasi* yang dilakukan sampai aplikasi web peduli sampah dan daur ulang Kota Yogyakarta selesai dan sesuai dengan kebutuhan Dinas Lingkungan Hidup Kota Yogyakarta dilakukan sebanyak 1 kali *iterasi*. Hasil yang ditampilkan pada penelitian ini merupakan *prototype iterasi* terakhir yang telah disetujui oleh pengguna yaitu Dinas Lingkungan Hidup Kota Yogyakarta.