

**IMPLEMENTASI METODE KLASIFIKASI FUZZY DALAM  
PENGELOMPOKAN *POSTING* GRUP LAPAK JUAL BELI  
PADA FACEBOOK UNTUK MEMPERMUDAH PENCARIAN  
INFORMASI BARANG**

**Nama Mahasiswa** : Helena Nadia Pratiwi  
**NRP** : 5110100155  
**Jurusan** : Teknik Informatika FTIf-ITS  
**Dosen Pembimbing I** : Dwi Sunaryono, S.Kom., M.Kom.  
**Dosen Pembimbing II** : Abdul Munif, S.Kom., M.Sc.

**ABSTRAK**

Saat ini media sosial, salah satunya Facebook telah digunakan untuk berbagai kepentingan salah satunya sebagai tempat jual beli barang. Salah satu fitur yang sering digunakan adalah fitur grup pada Facebook. Akan tetapi model timeline pada grup Facebook dimana posting akan ditampilkan berdasarkan urutan kemunculannya mengakibatkan kesulitan dalam hal mencari posting tertentu. Terlebih lagi jika posting tersebut sudah cukup lama maka tentu saja posting tersebut telah tertindih oleh posting-posting lain.

Untuk mengatasi masalah tersebut diperlukan metode untuk mengelompokkan data posting berdasarkan kategori. Metode pengelompokan yang digunakan adalah metode klasifikasi Fuzzy yang memungkinkan sebuah data untuk dapat memiliki lebih dari satu kelas dengan nilai keanggotaan yang berbeda. Dengan sebelumnya dilakukan pra-proses pada konten serta perhitungan bobot tiap kata menggunakan pembobotan TF-IDF, dapat dilakukan proses training data untuk mendapatkan model yang dapat digunakan untuk melakukan proses testing data untuk memberikan label kategori pada data. Aplikasi telah diuji untuk melakukan pengelompokan 100 data posting ke dalam 6 kategori dan memiliki tingkat ketelitian sebesar 0,76. Hasil tersebut

menunjukkan bahwa aplikasi dapat digunakan untuk mengelompokkan data posting berdasarkan kategori.

**Kata kunci: Facebook, fuzzy, grup, klasifikasi, posting.**

# **IMPLEMENTATION OF FUZZY CLASSIFICATION METHOD IN GROUPING SELL AND BUY STALLS GROUP'S POSTING ON FACEBOOK TO FACILITATE THE SEARCH OF GOODS INFORMATION**

**Name** : Helena Nadia Pratiwi  
**NRP** : 5110100155  
**Major** : Informatics Engineering, FTIf – ITS  
**Advisor I** : Dwi Sunaryono, S.Kom., M.Kom.  
**Advisor II** : Abdul Munif, S.Kom., M.Sc.

## **ABSTRACT**

Nowadays social media, such as Facebook has been used for many importances. One of them is as a place for sell and buy goods. One of the future that frequently used is group future on Facebook. However, on Facebook timeline postings are shown in order of their appearance. This kind of timeline makes it hard to find specific posting. Moreover if posting has been posted long enough, so it has been suppressed by many other postings and it is even harder to find that posting.

To overcome those problem, there is a need to find method that can group posting datas based on their category. Method that is used is Fuzzy classification method where a data can have more than one class with different membership values. After some pre-processing data on its content and weight calculation of each words in data using TF-IDF calculation, training data can be done in order to get models that can be used for purpose of testing data to give class label on data. Application has been tested to group 100 posting data into 6 categories and resulting accuracy value around 0,76. Those result proves that application can be used to group posting data based on its category.

**Keywords:** Classification, facebook, fuzzy, group, posting.

## DAFTAR KODE SUMBER

Kode Sumber 7.1 Kelas MySQLAccess .....	116
Kode Sumber 7.2 Kelas PostGrabber.....	118
Kode Sumber 7.3 Kelas Stemmer .....	120
Kode Sumber 7.4 Kelas DataPartition .....	121
Kode Sumber 7.5 Kelas Training .....	126
Kode Sumber 7.6 Kelas Testing .....	128
Kode Sumber 7.7 Fungsi HitungCosineSimilarity pada Kelas Testing.....	131
Kode Sumber 7.8 Kelas Navigation.....	135
Kode Sumber 7.9 Fungsi CariPost pada Kelas Cari .....	136
Kode Sumber 7.10 Fungsi PostIklan pada Kelas Write_wall....	138
Kode Sumber 7.11 Fungsi Like pada Kelas Mem_only .....	138
Kode Sumber 7.12 Fungsi Comment pada Kelas Mem_only ....	139
Kode Sumber 7.13 Fungsi Hapus pada Kelas Admin_only.....	139
Kode Sumber 7.14 Fungsi SetThres pada Kelas Admin_only ...	140

## **BAB II**

### **DASAR TEORI**

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan tugas akhir ini.

#### **2.1. Facebook API untuk Grup**

API (Application Programming Interface) atau antarmuka pemrograman aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API dapat menjelaskan cara sebuah tugas (task) tertentu dilakukan. Dalam pemrograman prosedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi yang disediakan[4].

Facebook API untuk grup memiliki sejumlah fungsi yang dapat digunakan agar aplikasi dapat berinteraksi dengan Facebook khususnya untuk fitur-fitur di dalam grup seperti mengakses anggota dari grup, mengakses posting dari grup, dan lain-lain.

#### **2.2. Facebook Graph API**

Facebook Graph API adalah tools pemrograman yang didesain untuk mendukung akses lebih ke ketentuan yang terdapat pada platform media sosial Facebook. Inti dari platform Facebook terletak pada social graph yang merupakan elemen yang bertanggung jawab untuk memfasilitasi semua hubungan online antar orang, tempat, benda, dan lain-lain. Facebook Graph API secara spesifik mendukung perluasan atau penggunaan graph oleh developer.

Bagian inti dari Facebook Graph API adalah daftar label untuk atribut objek yang berbeda. Dengan menggunakan standar yang telah disediakan ini, developer dapat memastikan bahwa perubahan yang mereka lakukan sesuai dengan syntax konvensional yang telah umum digunakan untuk halaman dan objek Facebook. Contohnya, API dapat melibatkan label seperti nama atau user yang

mengidentifikasi pihak tertentu atau label numerik untuk likes dan aktivitas kuantitatif lainnya, untuk tetap dapat memantau berapa banyak fungsi ini telah diterapkan pada entitas Facebook yang diberikan.

Social Graph API adalah cara otomatis untuk halaman-halaman web untuk mendeklarasikan hubungan sosial di antara mereka. Contohnya, banyak website yang memberikan link ke akun Twitter mereka. Di belakang layar, link tersebut dilakukan dengan cara yang mengizinkan developer dari produk sosial untuk mengetahui bahwa di sana terdapat hubungan sosial antara halaman web dengan akun Twitter[5].

### 2.3. Facebook Query Language

Facebook Query Language (FQL) adalah bahasa query yang memungkinkan programmer untuk dapat menggunakan SQL-style interface untuk melakukan query data yang didapatkan oleh Graph API. FQL menyediakan sejumlah fitur yang tidak tersedia di Graph API[6]. Data yang dikembalikan dari FQL query dapat dalam format XML maupun JSON (dapat diubah dengan menggunakan Format Query Parameter). Tidak seperti SQL, pada klausa FROM pada FQL hanya dapat mengandung satu buah tabel. Programmer dapat menggunakan klausa IN untuk SELECT atau klausa WHERE untuk melakukan subqueries, namun subqueries tidak dapat merujuk variabel yang berada di luar dari lingkup query[7].

Contoh dari FQL ditunjukkan pada Gambar 2.1.

```
SELECT status_id,message,TIME,SOURCE FROM `status` WHERE uid = me()
```

**Gambar 2.1 Contoh FQL**

### 2.4. JavaScript Object Notation (JSON)

JSON adalah format standar yang menggunakan human-readable text untuk melakukan transmisi objek data yang terdiri atas pasangan atribut-nilai. JSON biasanya digunakan untuk melakukan transmisi data antara server dan aplikasi web, sebagai alternatif dari XML. Meskipun awalnya JSON diturunkan dari JavaScript, JSON merupakan bahasa dengan format data yang independen dan bersifat

universal. Saat ini hampir semua bahasa pemrograman mendukung format JSON[8].

JSON dibangun dengan dua struktur, yaitu:

- ✚ Kumpulan pasangan atribut dan nilai. Dalam sejumlah bahasa, hal ini diasumsikan sebagai objek, record, struct, dictionary, hash table, keyed list, atau associative array.
- ✚ Daftar berurutan dengan nilai. Dalam sebagian besar bahasa, hal ini diasumsikan sebagai array, vector, list, atau sequence.

Pada JSON, sebuah objek merupakan himpunan tidak terstruktur dari pasangan atribut dan nilai. Deklarasi sebuah objek dimulai dengan ‘{’ (kurung kurawal buka) dan diakhiri dengan ‘}’ (kurung kurawal tutup). Setiap atribut diikuti oleh ‘:’ (tanda titik dua) dan setiap pasangan atribut dan nilai dipisahkan oleh ‘,’ (tanda koma). Sedangkan sebuah array merupakan kumpulan nilai yang terstruktur. Deklarasi array dimulai dengan ‘[’ (kurung siku buka) dan diakhiri dengan ‘]’ (kurung siku tutup). Setiap nilai dipisahkan oleh ‘,’ (tanda koma). Sebuah nilai dapat berupa string yang diapit oleh tanda petik ganda, angka, true atau false, nilai null, objek, atau array[9]. Gambar 2.2 menunjukkan contoh JSON.

## 2.5. Pembobotan TF-IDF

Pembobotan TF-IDF adalah metode yang digunakan dalam membuat definisi sebuah dokumen dalam bentuk model ruang vektor. Perhitungan TF-IDF menggabungkan dua buah perhitungan yaitu perhitungan term frequency (TF) dan perhitungan inverse document frequency (IDF)[10].

Cara perhitungan TF-IDF adalah sebagai berikut: Misalkan  $n$  adalah jumlah dokumen yang terkumpul dan kata kunci  $k_i$  muncul pada sejumlah  $n_i$  dokumen. Kemudian misalkan  $f_{i,j}$  adalah banyaknya kata kunci  $k_i$  yang ditemukan pada dokumen  $d_j$ . Maka TF, yaitu term frequency dari kata kunci  $k_i$  pada dokumen  $d_j$  dapat didefinisikan pada Persamaan 2.1.

$$TF = \frac{f_{i,j}}{\text{length}(d_j)} \quad (2.1)$$

```

{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "height_cm": 167.64,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ]
}

```

**Gambar 2.2 Contoh JSON**

Karena tidak semua kata kunci yang muncul pada tiap dokumen bisa digunakan untuk memisahkan antara kata-kata yang benar-benar relevan dengan yang tidak, maka digunakanlah perhitungan Inverse Document Frequency (IDF) yang nantinya digabungkan dengan perhitungan TF. Nilai IDF dari sebuah kata kunci  $k_i$  dapat didefinisikan pada Persamaan 2.2.

$$IDF = \log \frac{n}{n_i} \quad (2.2)$$

Kemudian, bobot TF-IDF dari sebuah kata kunci  $k_i$  pada dokumen  $d_j$  dapat didefinisikan pada Persamaan 2.3.

$$w_{i,j} = TF_{i,j} \times IDF_i \quad (2.3)$$

Setelah pembobotan dilakukan pada tiap kata kunci, konten dari sebuah dokumen yang direpresentasikan dalam bentuk model ruang vektor dapat didefinisikan pada Persamaan 2.4.

$$content(d_j) = \langle w_{1j}, w_{2j}, \dots, w_{kj} \rangle \quad (2.4)$$

Pembobotan TF-IDF merefleksikan seberapa penting sebuah kata pada dokumen dalam kumpulan dokumen. TF-IDF sering



digunakan sebagai faktor bobot dalam pencarian informasi dan text mining. Variasi dari pembobotan TF-IDF sering digunakan oleh mesin pencarian sebagai media utama dalam menghitung dan memberi peringkat terhadap tingkat relevansi dari dokumen berdasarkan query pengguna. TF-IDF dapat digunakan untuk menyaring stop-words dalam berbagai bidang subjek termasuk peringkasan teks dan klasifikasi.

## **2.6. Klasifikasi Fuzzy**

Klasifikasi Fuzzy adalah metode klasifikasi data yang mengizinkan atribut-atribut data untuk menerapkan nilai keanggotaan pada objek sehingga sebuah objek dapat dianggap sebagai anggota parsial dari sebuah kelas. Keanggotaan kelas biasanya didefinisikan dengan skala kontinyu dari nol hingga satu dimana nol berarti objek bukan merupakan anggota dari kelas dan satu berarti objek memiliki keanggotaan penuh pada kelas tersebut[11]. Dengan menggunakan klasifikasi Fuzzy, sebuah data dapat memiliki lebih dari satu kelas dengan nilai keanggotaan yang berbeda-beda. Contoh posting yang memiliki lebih dari satu kelas ditunjukkan pada Gambar 2.3. Seperti ditunjukkan pada Gambar 2.3, pada klasifikasi standar data posting tersebut hanya akan memiliki satu kelas yaitu kelas jasa. Sedangkan jika menggunakan klasifikasi Fuzzy maka data posting tersebut dapat dimasukkan dalam kelas jasa sekaligus kelas elektronik dengan nilai keanggotaan yang berbeda.

Klasifikasi termasuk ke dalam area pengenalan pola dan mesin pembelajaran. Sifat dari klasifikasi Fuzzy adalah sebagai berikut:

### **1. Soft labelling**

Asumsi standar dari pengenalan pola adalah bahwa kelas-kelas yang ada bersifat mutually exclusive atau saling lepas. Hal ini tidak berlaku pada klasifikasi Fuzzy. Seperti ditunjukkan pada contoh pada Gambar 2.3, klasifikasi Fuzzy dapat memberikan derajat keanggotaan (soft labels) pada kelas jasa dan kelas elektronik sedangkan klasifikasi standar hanya dapat memberikan satu label kelas saja yaitu kelas jasa.

WTS

Printer anda bermasalah?  
tak usah GALAU mai bro...  
Cepet aja kontak kami 085852004947 atau 031-34446661, SMS juga boleh...  
SPECIALIS PRINTER SAKINAH SUPERMARKET  
buka dari pukul 08.00 sd 21.00  
Melayani :

1. Servis Printer
2. Refil tinta/toner
3. Modifikasi printer

GRATIS LAYANAN JEMPUT ANTAR LHO...  
HARI AHAD ATAU TANGGAL MERAH TETAP BUKA CIHUIII

**Gambar 2.3 Contoh Posting yang Memiliki Lebih dari Satu Kelas**

## 2. Interpretability

Klasifikasi Fuzzy seringkali didesain transparan yang berarti bahwa tahapan dan pernyataan logika yang mengarah ke proses memprediksi kelas dapat dilacak dan dipahami.

## 3. Data terbatas dan memerlukan keahlian khusus

Tidak semua data dapat diolah dengan menggunakan klasifikasi Fuzzy. Contoh data yang dapat diolah dengan klasifikasi Fuzzy antara lain data penyakit langka, aktivitas teroris, dan bencana alam. Klasifier Fuzzy dapat dibangun dengan menggunakan opini dari pakar, data, atau keduanya.

Dalam pengerjaan tugas akhir ini, metode klasifikasi Fuzzy yang digunakan menerapkan konsep K-Means. Dalam statistik dan mesin pembelajaran, pengelompokan K-Means merupakan metode analisis kelompok yang mengarah pada pemartisian  $N$  objek pengamatan ke dalam  $K$  kelompok (cluster) dimana setiap objek pengamatan dimiliki oleh sebuah kelompok dengan mean (rata-rata) terdekat, mirip dengan algoritma Expectation-Maximization untuk Gaussian Mixture di mana keduanya mencoba untuk menemukan pusat dari kelompok dalam data sebanyak iterasi perbaikan yang

dilakukan oleh kedua algoritma[12]. Aplikasi yang dibangun akan menggunakan metode klasifikasi untuk membangun model yang nantinya akan digunakan untuk menentukan kelas dari data. Sedangkan konsep K-Means akan digunakan untuk mengukur tingkat kedekatan sebuah data dengan kelompok tertentu pada model. Nilai tingkat kedekatan sebuah data dengan kelompok pada model tersebut akan digunakan sebagai derajat keanggotaan data dalam kelompok sehingga sebuah data dapat dikelompokkan ke dalam lebih dari satu kategori tertentu dilihat dari derajat keanggotaan data tersebut.

## **2.7. K-Fold Cross Validation**

Cross Validation atau disebut juga rotation estimation adalah teknik validasi model untuk menilai bagaimana hasil analisis statistik dapat melakukan pengelompokan pada himpunan data yang independen. Teknik ini sering digunakan dalam mengestimasi seberapa akurat sebuah model dalam melakukan pengelompokan. Teknik ini memisahkan sebuah himpunan data menjadi data training dan data validasi. Data training digunakan untuk membangun model sedangkan data validasi digunakan untuk menguji model yang telah dibuat dari tahap training. K-Fold Cross Validation merupakan salah satu jenis dari Cross Validation.

Dalam K-Fold Cross Validation, himpunan data secara acak dibagi menjadi k bagian atau fold. Dari k bagian tersebut, satu bagian digunakan sebagai data validasi untuk menguji model sedangkan k-1 bagian lainnya digunakan sebagai data training. Proses Cross Validation kemudian diulang sebanyak k kali dengan setiap bagian data digunakan sebagai data validasi sebanyak satu kali. Hasil iterasi sebanyak k kali tersebut kemudian dirata-rata untuk mendapatkan sebuah estimasi.

Tujuan dari Cross Validation adalah untuk mengurangi masalah seperti overfitting model serta mampu memberikan gambaran bagaimana model mampu mengelompokkan himpunan data yang independen[13].

## 2.8. Cosine Similarity

Cosine Similarity adalah metode yang dapat digunakan untuk mencari tingkat kemiripan dari dua buah dokumen yang direpresentasikan dalam vektor bobot TF-IDF. Pengukuran cosine similarity dari dua buah vektor dapat didefinisikan pada Persamaan 2.5.

$$\begin{aligned} \cos(\vec{w}_c, \vec{w}_u) &= \frac{\vec{w}_c \cdot \vec{w}_u}{\|\vec{w}_c\|_2 \times \|\vec{w}_u\|_2} \\ &= \frac{\sum_{i=1}^K w_{c,i} w_{u,i}}{\sqrt{\sum_{i=1}^K w_{c,i}^2} \sqrt{\sum_{i=1}^K w_{u,i}^2}} \end{aligned} \quad (2.5)$$

## 2.9. Presisi dan Recall

Untuk menghitung tingkat performansi suatu sistem yang memiliki kemampuan untuk me-retrieve informasi-informasi tertentu dapat digunakan perhitungan presisi dan recall. Presisi adalah probabilitas informasi yang relevan dari semua informasi yang di-retrieve sistem. Secara matematis, rumus untuk menghitung presisi dapat dilihat pada Persamaan 2.6.

$$precision = \frac{\text{relevant items retrieved}}{\text{retrieved items}} \quad (2.6)$$

Recall adalah probabilitas informasi relevan yang di-retrieve dibandingkan jumlah informasi-informasi yang relevan. Secara matematis rumus untuk menghitung recall dapat dilihat pada Persamaan 2.7.

$$recall = \frac{\text{relevant items retrieved}}{\text{relevant items}} \quad (2.7)$$

## 2.10. F Measure

F Measure merupakan salah satu perhitungan evaluasi dalam temu kembali informasi yang mengkombinasikan nilai presisi dan recall. Nilai presisi dan recall pada suatu keadaan dapat memiliki

bobot yang berbeda. Ukuran yang menampilkan timbal balik antara presisi dan recall adalah F Measure yang merupakan bobot harmonic mean dari nilai presisi dan recall. Secara matematis, rumus untuk menghitung nilai F Measure atau  $F_1$  ditunjukkan pada Persamaan 2.8.

$$F_1 = 2 \times \frac{\textit{presisi} \times \textit{recall}}{\textit{presisi} + \textit{recall}} \quad (2.8)$$

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML (Unified Modelling Language).

#### **3.1. Analisis**

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

##### **3.1.1. Analisis Permasalahan**

Facebook sebagai media sosial yang populer di masyarakat telah dimanfaatkan oleh banyak pihak untuk berbagai kepentingan salah satunya sebagai tempat jual beli barang secara online. Dengan fitur grup pada Facebook, pengguna dapat berbagi informasi mengenai barang maupun jasa yang sedang ditawarkan maupun dibutuhkan dengan pengguna lain yang juga merupakan anggota dari grup tersebut. Akan tetapi, model timeline di Facebook yang berupa sebuah streamline seringkali membuat pengguna kesulitan dalam mencari informasi tertentu karena posting akan ditampilkan berdasarkan urutan kemunculannya. Sebagai akibatnya, hal ini tentu mempersulit pencarian informasi yang sudah cukup lama karena telah tertumpuk oleh posting-posting lain yang lebih baru.

Oleh karena itu, aplikasi ini dibuat agar pengguna Facebook lebih khususnya anggota dari grup Lapak Jual Beli ITS lebih mudah dalam mencari informasi barang atau jasa tertentu karena informasi tersebut dalam hal ini posting di grup Facebook telah dikelompokkan berdasarkan kategori terkait dari barang atau jasa yang ditawarkan

atau dibutuhkan pada posting tersebut. Selain mempermudah pencarian informasi, diharapkan aplikasi ini juga membantu pihak berkepentingan dalam grup (pihak yang ingin membeli atau ingin menawarkan barang atau jasa) dalam menyampaikan kepentingan mereka.

### **3.1.2. Deskripsi Umum Sistem**

Aplikasi yang dibangun pada tugas akhir ini adalah aplikasi web yang melakukan pengelompokan terhadap posting di grup Lapak Jual Beli ITS di Facebook. Data posting, informasi grup, dan informasi mengenai anggota grup didapatkan dari Facebook API yang kemudian akan dikelompokkan berdasarkan kategori terkait dari tiap posting dengan menggunakan metode klasifikasi Fuzzy. Keluaran yang dihasilkan adalah posting grup Lapak Jual Beli ITS yang telah terkelompokkan berdasarkan kategori terkait.

Selain melakukan pengelompokan posting, aplikasi ini juga dapat menjembatani pengguna dengan grup tempat jual beli. Aplikasi ini menyediakan fitur login dengan akun Facebook sehingga pengguna dapat berpartisipasi secara aktif di dalam grup seperti mem-posting iklan, memberikan komentar, atau memberikan like pada posting tertentu.

Dengan adanya aplikasi ini diharapkan dapat mempermudah pengguna dalam melakukan pencarian informasi barang atau jasa tertentu yang terdapat di grup Lapak Jual Beli ITS. Selain itu juga diharapkan aplikasi ini membantu proses tersampainya kepentingan para pengguna di dalam grup.

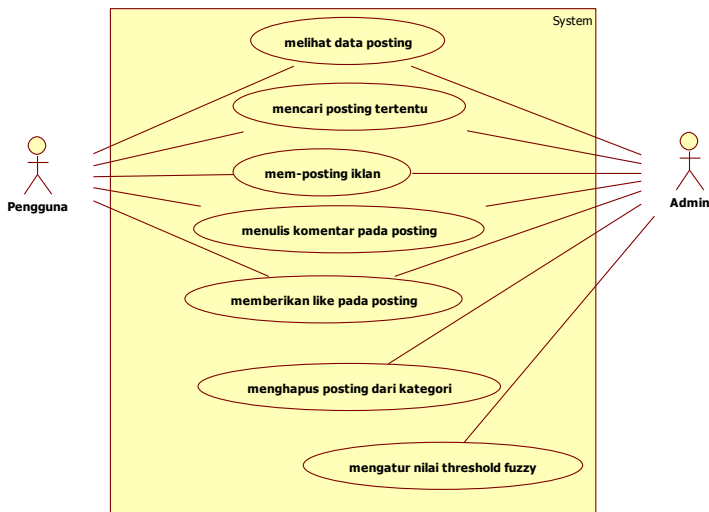
### **3.1.3. Aktor**

Terdapat dua jenis aktor dari aplikasi ini, yaitu pengguna dan admin aplikasi. Pengguna aplikasi dapat melihat data posting yang telah dikelompokkan, melakukan pencarian data posting tertentu, mem-posting iklan, memberikan komentar pada posting, dan memberikan like pada posting. Untuk dapat mem-posting iklan, memberi komentar, dan memberi like pada posting pengguna harus login dengan akun Facebook terlebih dahulu. Sedangkan untuk admin aplikasi selain memiliki hak akses yang dimiliki pengguna,

juga memiliki hak akses tambahan yaitu menghapus posting yang tidak sesuai dari kategori dan juga mengatur nilai threshold Fuzzy.

### 3.1.4. Kasus Penggunaan

Berdasarkan analisis aktor yang telah dijelaskan pada subbab sebelumnya, maka dibuat diagram kasus penggunaan yang menjelaskan fungsionalitas perangkat lunak yang dikembangkan. Kasus penggunaan dari perangkat lunak ini digambarkan oleh diagram kasus penggunaan pada Gambar 3.1. Tabel 3.1 menjelaskan masing-masing kasus penggunaan.



**Gambar 3.1 Diagram Kasus Penggunaan**

**Tabel 3.1 Deskripsi Kasus Penggunaan**

No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1.	UC-0001	Melihat data posting	Pengguna dapat melihat data posting yang telah terkelompokkan berdasarkan kategori.



No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
2.	UC-0002	Mencari posting tertentu	Pengguna dapat mencari posting tertentu dengan menggunakan fitur search pada aplikasi.
3.	UC-0003	Mem-posting iklan	Pengguna dapat mem-posting konten berupa iklan ke grup melalui aplikasi.
4.	UC-0004	Menulis komentar pada posting	Pengguna dapat memberikan komentar pada posting tertentu melalui aplikasi.
5.	UC-0005	Memberikan like pada posting	Pengguna dapat memberikan like pada posting tertentu melalui aplikasi.
6.	UC-0006	Menghapus posting dari kategori	Admin dapat menghapus posting yang dianggap kurang relevan dari kategori.
7.	UC-0007	Mengatur nilai threshold Fuzzy	Admin dapat mengatur derajat nilai keanggotaan data yang masih diterima oleh sistem.

#### 3.1.4.1. Melihat Data Posting

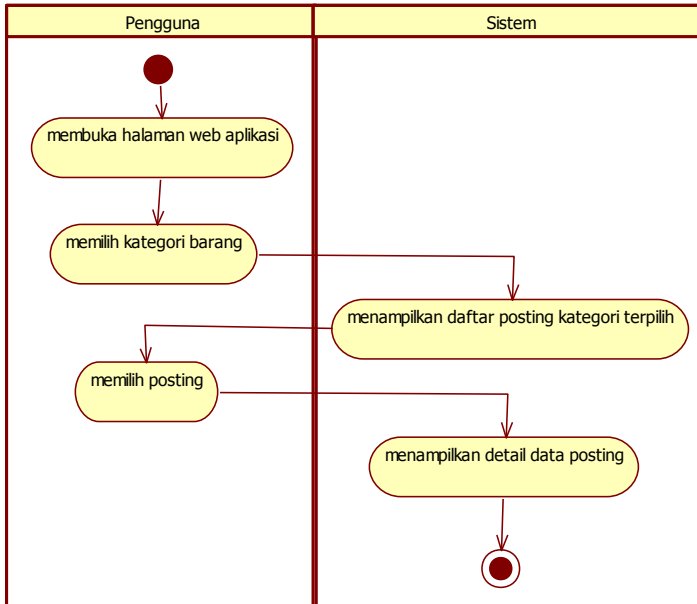
Fitur utama dari aplikasi ini adalah melihat data posting yang telah dikelompokkan berdasarkan kategori. Spesifikasi kasus penggunaan melihat data posting dapat dilihat pada Tabel 3.2. Diagram aktivitas dari aplikasi ini dapat dilihat pada Gambar 3.2.

**Tabel 3.2 Spesifikasi Kasus Penggunaan: Melihat Data Posting**

<b>Nama Kasus Penggunaan</b>	Melihat Data Posting
<b>Nomor</b>	UC-0001
Deskripsi	Pengguna dapat melihat data posting yang telah dikelompokkan berdasarkan kategori.
Aktor	Pengguna aplikasi.
Kondisi Awal	Pengguna telah membuka halaman web aplikasi.
Alur Normal	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman web dari aplikasi.</li> <li>2. Pengguna memilih kategori barang yang ingin dilihat.</li> <li>3. Sistem menampilkan daftar posting dari kategori yang telah dipilih.</li> <li>4. Pengguna memilih salah satu posting.</li> <li>5. Sistem menampilkan detail data dari posting.</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Data barang dengan kategori terpilih ditampilkan di halaman web aplikasi.

### 3.1.4.2. Mencari Posting Tertentu

Dalam aplikasi ini, terdapat fitur search untuk mencari posting tertentu berdasarkan kata kunci yang dimasukkan oleh pengguna. Spesifikasi kasus penggunaan mencari posting tertentu dijelaskan pada Tabel 3.3. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.3.



**Gambar 3.2 Diagram Aktivitas Melihat Data Posting**

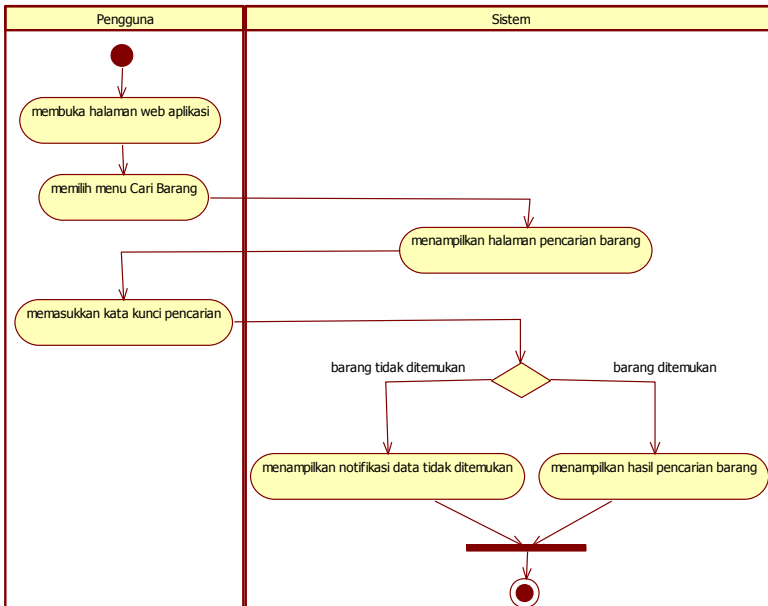
**Tabel 3.3 Spesifikasi Kasus Penggunaan: Mencari Posting Tertentu**

<b>Nama Kasus Penggunaan</b>	Mencari Posting Tertentu
<b>Nomor</b>	UC-0002
<b>Deskripsi</b>	Pengguna dapat mencari posting tertentu dengan menggunakan fitur search pada aplikasi.
<b>Aktor</b>	Pengguna aplikasi.
<b>Kondisi Awal</b>	Pengguna telah membuka halaman web aplikasi.

<b>Nama Kasus Penggunaan</b>	Mencari Posting Tertentu
Alur Normal	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman web dari aplikasi.</li> <li>2. Pengguna memilih menu Cari Barang</li> <li>3. Sistem menampilkan halaman pencarian barang.</li> <li>4. Pengguna memasukkan kata kunci barang yang ingin dicari.</li> <li>5. Sistem menampilkan hasil pencarian barang yang bersesuaian ke halaman web.               <ol style="list-style-type: none"> <li>A1. Barang yang dicari tidak ditemukan.</li> </ol> </li> </ol>
Alur Alternatif	<ol style="list-style-type: none"> <li>A1. Barang yang dicari tidak ditemukan.               <ol style="list-style-type: none"> <li>1. Sistem menampilkan notifikasi bahwa barang tidak ditemukan.</li> <li>2. Selesai.</li> </ol> </li> </ol>
Kondisi Akhir	Data hasil pencarian ditampilkan di panel hasil pencarian aplikasi.

### 3.1.4.3. Mem-posting Iklan

Melalui aplikasi ini, pengguna dapat mem-posting iklan yang nantinya akan terkirim secara otomatis ke wall dari grup Lapak Jual Beli ITS di Facebook. Untuk dapat mem-posting iklan, terlebih dahulu pengguna harus login dengan akun Facebook. Spesifikasi kasus penggunaan mem-posting iklan dijelaskan pada Tabel 3.4. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.4.



**Gambar 3.3 Diagram Aktivitas Mencari Posting Tertentu**

**Tabel 3.4 Spesifikasi Kasus Penggunaan: Mem-posting Iklan**

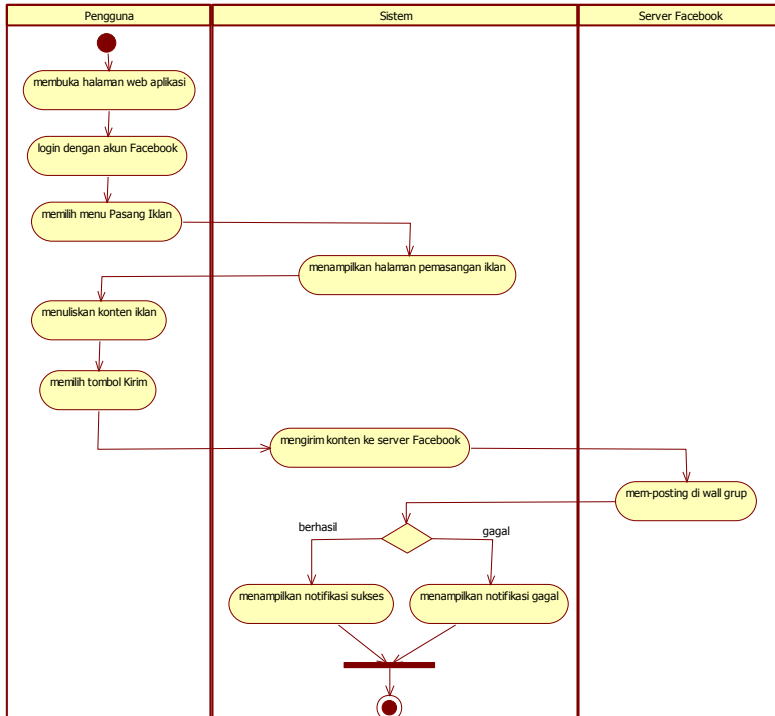
<b>Nama Kasus Penggunaan</b>	Mem-posting Iklan
<b>Nomor</b>	UC-0003
<b>Deskripsi</b>	Pengguna dapat mem-posting konten berupa iklan ke grup melalui aplikasi.
<b>Aktor</b>	Pengguna aplikasi yang melakukan login dengan akun Facebook dan merupakan anggota dari grup.
<b>Kondisi Awal</b>	Pengguna telah membuka halaman web aplikasi.
<b>Alur Normal</b>	1. Pengguna membuka halaman web dari

<b>Nama Kasus Penggunaan</b>	Mem-posting Iklan
	<p>aplikasi.</p> <ol style="list-style-type: none"> <li>2. Pengguna melakukan login dengan akun Facebook mereka.</li> <li>3. Pengguna memilih menu Pasang Iklan</li> <li>4. Sistem menampilkan halaman pemasangan iklan.</li> <li>5. Pengguna menulis konten iklan di form pemasangan iklan.</li> <li>6. Pengguna memilih tombol 'kirim'.</li> <li>7. Sistem mengirim konten ke server Facebook.</li> <li>8. Server Facebook menerima konten yang dikirim sistem dan mem-posting di wall grup.</li> <li>9. Sistem menampilkan notifikasi bahwa konten berhasil di-posting.</li> </ol> <p>A1. Konten gagal di-posting.</p>
Alur Alternatif	<p>A1. Konten gagal di-posting</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan notifikasi bahwa konten gagal di-posting.</li> <li>2. Selesai.</li> </ol>
Kondisi Akhir	Konten iklan berhasil di-posting.

#### 3.1.4.4. Menulis Komentar pada Posting

Dalam aplikasi ini, pengguna dapat memberikan komentar pada data posting yang tersedia. Akan tetapi hanya pengguna yang merupakan anggota dari grup saja yang dapat memberi komentar pada posting. Komentar tersebut secara otomatis akan ditambahkan

pada data posting yang bersangkutan. Spesifikasi kasus penggunaan ini dijelaskan pada Tabel 3.5. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.5.



**Gambar 3.4 Diagram Aktivitas Mem-posting Iklan**

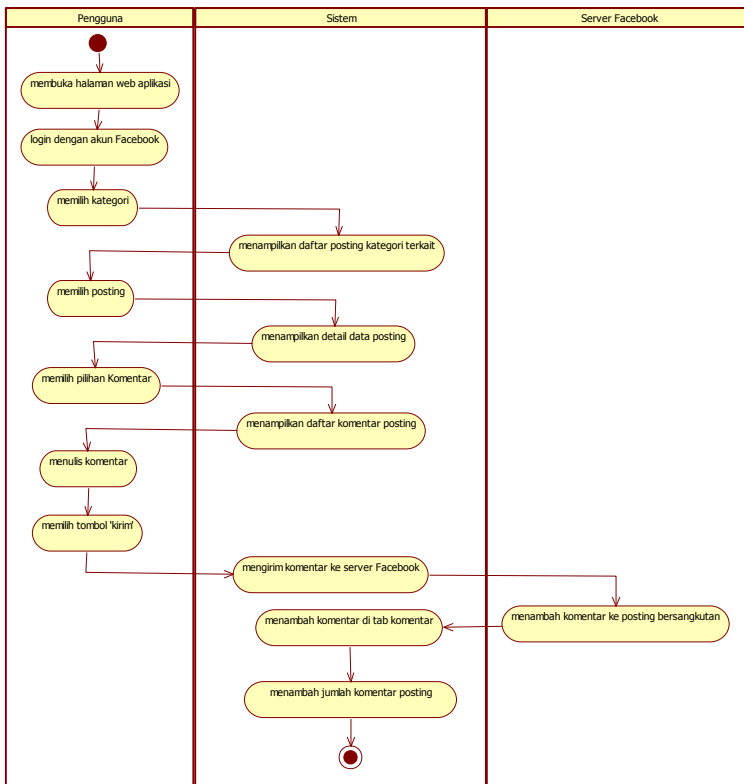
**Tabel 3.5 Spesifikasi Kasus Penggunaan: Menulis Komentar pada Posting**

<b>Nama Kasus Penggunaan</b>	Menulis Komentar pada Posting
<b>Nomor</b>	UC-0004
<b>Deskripsi</b>	Pengguna dapat memberikan komentar pada posting tertentu melalui aplikasi.

<b>Nama Kasus Penggunaan</b>	Menulis Komentar pada Posting
Aktor	Pengguna aplikasi yang melakukan login dengan akun Facebook dan merupakan anggota grup jual beli di Facebook.
Kondisi Awal	Pengguna telah membuka halaman web aplikasi.
Alur Normal	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman web dari aplikasi.</li> <li>2. Pengguna melakukan login dengan akun Facebook mereka.</li> <li>3. Pengguna memilih kategori barang yang diinginkan.</li> <li>4. Sistem menampilkan daftar posting kategori terpilih.</li> <li>5. Pengguna memilih posting yang ingin diberi komentar.</li> <li>6. Sistem menampilkan detail data posting.</li> <li>7. Pengguna memilih pilihan 'Komentar' untuk membuka tab komentar dari posting.</li> <li>8. Sistem menampilkan daftar komentar dari posting pada tab komentar.</li> <li>9. Pengguna menulis komentar pada tab komentar posting bersangkutan.</li> <li>10. Pengguna menekan tombol 'kirim'.</li> <li>11. Sistem mengirim komentar ke server Facebook.</li> <li>12. Server Facebook menerima komentar yang dikirim sistem dan menambahkannya di tab komentar posting bersangkutan.</li> <li>13. Sistem menampilkan komentar pengguna pada tab komentar posting bersangkutan</li> </ol>



<b>Nama Kasus Penggunaan</b>	Menulis Komentar pada Posting
	dan menambahkan jumlah komentar dari posting.
Alur Alternatif	-
Kondisi Akhir	Komentar pengguna berhasil dikirim.



**Gambar 3.5** Diagram Aktivitas Menulis Komentar pada Posting

### 3.1.4.5. Memberikan Like pada Posting

Dalam aplikasi ini, pengguna dapat memberikan like pada data posting yang tersedia. Untuk dapat memberikan like pada posting, pengguna harus login dengan akun Facebook. Meskipun pengguna bukan merupakan anggota dari grup, pengguna tetap dapat memberikan like pada posting dari grup. Request like tersebut secara otomatis akan ditambahkan pada data posting yang bersangkutan. Spesifikasi kasus penggunaan ini dijelaskan pada Tabel 3.6. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.6.

**Tabel 3.6 Spesifikasi Kasus Penggunaan: Memberikan Like pada Posting**

<b>Nama Kasus Penggunaan</b>	Memberikan Like pada Posting
<b>Nomor</b>	UC-0005
<b>Deskripsi</b>	Pengguna dapat memberikan like pada posting tertentu melalui aplikasi.
<b>Aktor</b>	Pengguna aplikasi yang melakukan login dengan akun Facebook.
<b>Kondisi Awal</b>	Pengguna telah membuka halaman web aplikasi.
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman web dari aplikasi.</li> <li>2. Pengguna melakukan login dengan akun Facebook mereka.</li> <li>3. Pengguna memilih kategori barang yang diinginkan.</li> <li>4. Sistem menampilkan daftar posting kategori terpilih.</li> <li>5. Pengguna memilih posting yang ingin diberi like.</li> <li>6. Sistem menampilkan detail data posting terpilih.</li> <li>7. Pengguna memilih tombol 'Suka'.</li> </ol>

<b>Nama Kasus Penggunaan</b>	Memberikan Like pada Posting
	8. Sistem mengirim perintah like ke server Facebook. 9. Server Facebook menerima perintah like dan menambahkannya di tab like dari posting bersangkutan. 10. Sistem menambahkan jumlah suka pada posting bersangkutan. 11. Sistem mengubah status posting menjadi 'Disukai'.
Alur Alternatif	-
Kondisi Akhir	Perintah suka berhasil dikirim.

### 3.1.4.6. Menghapus Posting dari Kategori

Admin aplikasi memiliki hak akses untuk menghapus posting yang dianggap tidak terkelompokkan pada kategori yang sesuai dari kategori. Spesifikasi kasus penggunaan ini dijelaskan pada Tabel 3.7. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.7.

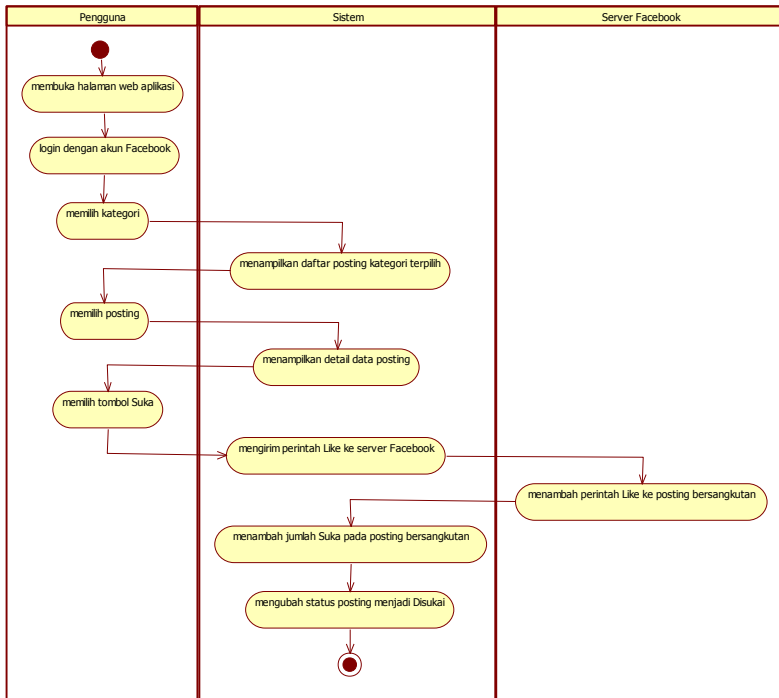
**Tabel 3.7 Spesifikasi Kasus Penggunaan: Menghapus Posting dari Kategori**

<b>Nama Kasus Penggunaan</b>	Menghapus Posting dari Kategori
<b>Nomor</b>	UC-0006
Deskripsi	Admin dapat menghapus posting dari kategori tertentu pada aplikasi.
Aktor	Admin aplikasi.
Kondisi	Admin telah membuka halaman web aplikasi.

<b>Nama Kasus Penggunaan</b>	<b>Menghapus Posting dari Kategori</b>
Awal	
Alur Normal	<ol style="list-style-type: none"> <li>1. Admin membuka halaman web dari aplikasi.</li> <li>2. Admin melakukan login dengan akun Facebook mereka.</li> <li>3. Admin memilih menu Dashboard.</li> <li>4. Sistem menampilkan halaman Dashboard menu Pengaturan Data.</li> <li>5. Admin memilih kategori barang yang diinginkan.</li> <li>6. Sistem menampilkan data posting yang termasuk dalam kategori terpilih.</li> <li>7. Admin memilih posting yang ingin dihapus dari kategori.</li> <li>8. Admin memilih tombol hapus.</li> <li>9. Sistem menghapus posting dari kategori terpilih.</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Posting berhasil dihapus dari kategori.

### 3.1.4.7. Mengatur Nilai Threshold Fuzzy

Admin aplikasi memiliki hak akses untuk mengatur nilai threshold Fuzzy yaitu derajat nilai keanggotaan data yang masih diterima oleh sistem. Spesifikasi kasus penggunaan ini dijelaskan pada Tabel 3.8. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.8.

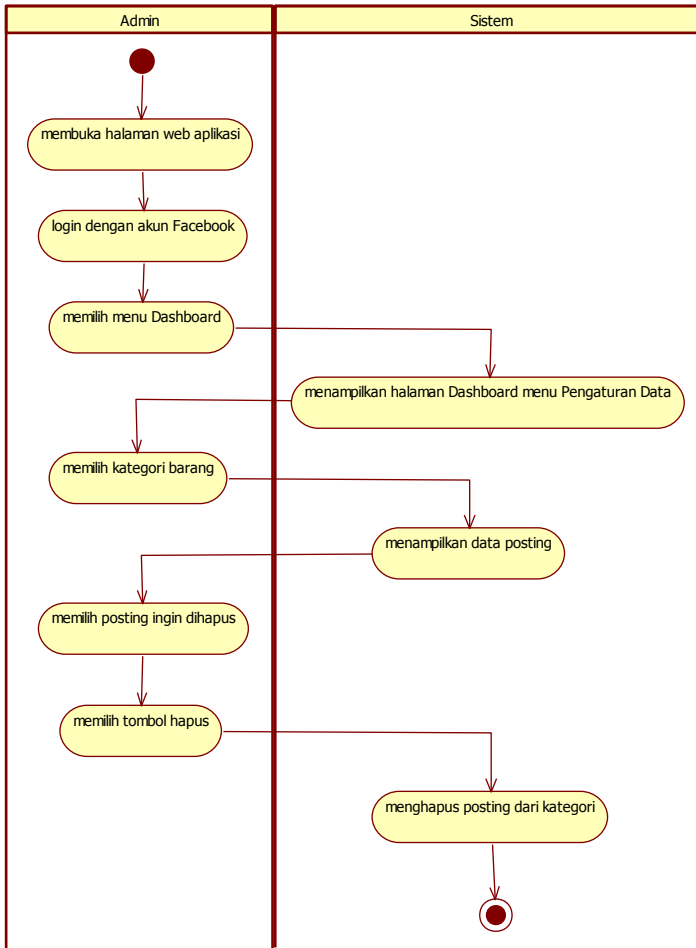


**Gambar 3.6 Diagram Aktivitas Memberikan Like pada Posting**

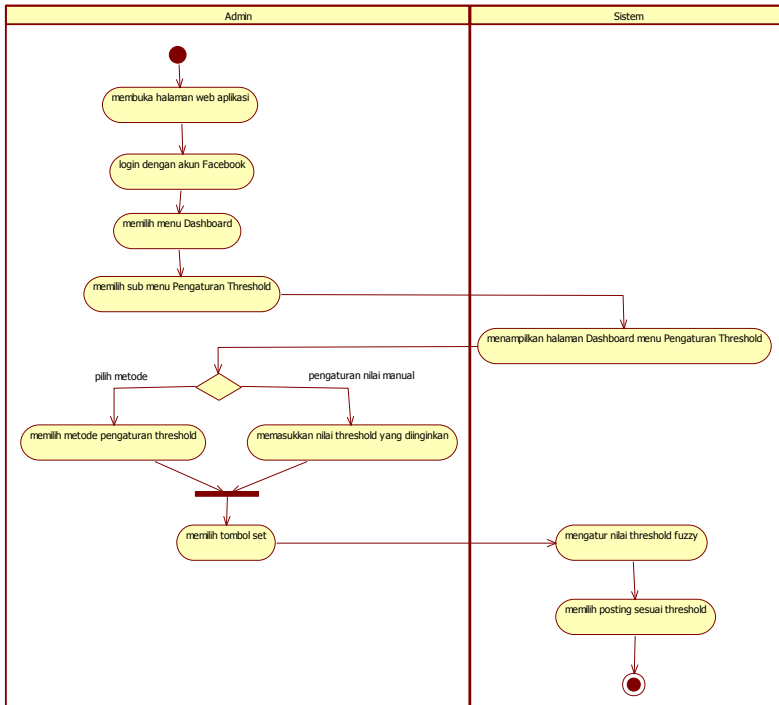
**Tabel 3.8 Spesifikasi Kasus Penggunaan: Mengatur Nilai Threshold Fuzzy**

<b>Nama Kasus Penggunaan</b>	Mengatur Nilai Threshold Fuzzy
<b>Nomor</b>	UC-0007
<b>Deskripsi</b>	Admin dapat mengatur nilai threshold Fuzzy dari aplikasi.
<b>Aktor</b>	Admin aplikasi.
<b>Kondisi Awal</b>	Admin telah membuka halaman web aplikasi.
<b>Alur Normal</b>	1. Admin membuka halaman web dari

<b>Nama Kasus Penggunaan</b>	Mengatur Nilai Threshold Fuzzy
	<p>aplikasi.</p> <ol style="list-style-type: none"> <li>2. Admin melakukan login dengan akun Facebook mereka.</li> <li>3. Admin memilih menu Dashboard.</li> <li>4. Admin memilih sub menu Pengaturan Threshold.</li> <li>5. Sistem menampilkan halaman Dashboard menu Pengaturan Threshold.</li> <li>6. Admin memilih metode pengaturan nilai threshold. <ol style="list-style-type: none"> <li>A1. Admin mengatur nilai threshold Fuzzy secara manual</li> </ol> </li> <li>7. Admin memilih tombol 'set'.</li> <li>8. Sistem mengatur nilai threshold Fuzzy.</li> <li>9. Sistem memilih posting yang termasuk dalam jangkauan dari threshold.</li> </ol>
Alur Alternatif	<ol style="list-style-type: none"> <li>A1. Admin mengatur nilai threshold Fuzzy secara manual <ol style="list-style-type: none"> <li>1. Admin memasukkan nilai threshold Fuzzy yang diinginkan.</li> <li>2. Kembali ke alur normal nomor 7.</li> <li>3. Selesai.</li> </ol> </li> </ol>
Kondisi Akhir	Nilai threshold Fuzzy berhasil diatur.



**Gambar 3.7 Diagram Aktivitas Menghapus Posting dari Kategori**

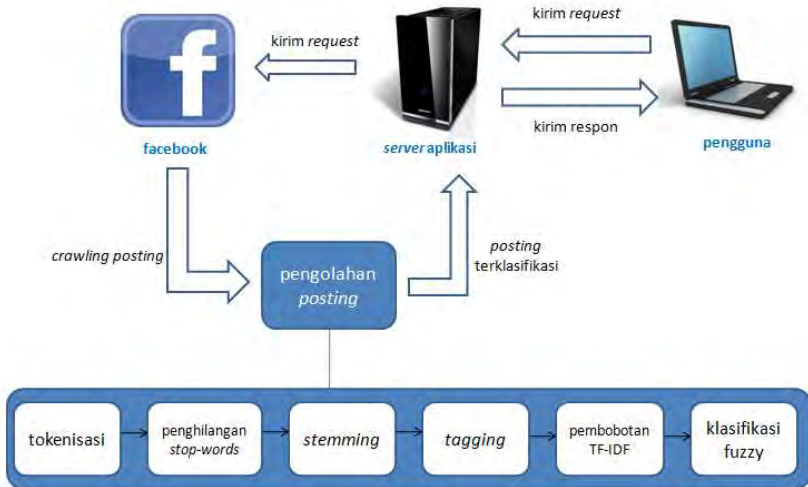


**Gambar 3.8 Diagram Aktivitas Mengatur Nilai Threshold Fuzzy**

### 3.1.5. Spesifikasi Kebutuhan Perangkat Lunak

Bagian ini berisi semua kebutuhan perangkat lunak yang diuraikan secara rinci dalam bentuk diagram kasus dan diagram aktivitas. Masing-masing diagram menjelaskan perilaku atau sifat dari sistem ini. Kebutuhan perangkat lunak dalam sistem ini mencakup kebutuhan fungsional saja. Pada bab ini juga dijelaskan tentang spesifikasi terperinci pada masing-masing kebutuhan fungsional. Rincian spesifikasi dari kasus penggunaan disajikan dalam bentuk tabel. Diagram arsitektur dari sistem digambarkan pada Gambar 3.9.





**Gambar 3.9** Arsitektur Perangkat Lunak

### 3.1.5.1. Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang harus dimiliki sistem. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan dan reaksi sistem terhadap masukan atau pada situasi tertentu. Daftar kebutuhan fungsional dapat dilihat pada Tabel 3.9.

**Tabel 3.9** Daftar Kebutuhan Fungsional Perangkat Lunak

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Mengelompokkan posting posting berdasarkan kategori.	Sistem menerima input berupa data posting dari server Facebook dan menghasilkan keluaran berupa data posting yang telah terkelompok berdasar kategori.
F-0002	Melakukan login dengan akun Facebook.	Sistem menerima input berupa alamat email dan password pengguna dan akan mengarahkan pengguna ke halaman anggota jika login berhasil.
F-0003	Mem-posting iklan.	Sistem menerima input berupa konten iklan yang akan di-posting dan akan mem-posting iklan ke wall grup melalui aplikasi.

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0004	Mencari posting tertentu.	Sistem menerima input berupa kata kunci pencarian dan akan menampilkan keluaran berupa data posting yang bersesuaian dengan kata kunci.
F-0005	Memberikan komentar pada posting.	Sistem menerima input berupa konten komentar dan akan menambahkan komentar dari posting yang bersangkutan.
F-0006	Memberikan like pada posting.	Sistem menerima input berupa perintah like dari pengguna dan akan menambahkan like tersebut ke posting yang bersangkutan.
F-0007	Menghapus posting dari kategori	Sistem menerima input berupa perintah penghapusan dari admin dan akan mengubah nilai keanggotaan dari posting pada kategori terpilih menjadi 0 dan secara tidak langsung menghapus posting dari kategori.
F-0008	Mengatur nilai threshold Fuzzy	Sistem menerima input berupa mode pengaturan nilai threshold Fuzzy dan akan memberikan flag pada posting dengan nilai keanggotaan di atas threshold untuk ditampilkan ke sistem.

## 3.2. Perancangan Sistem

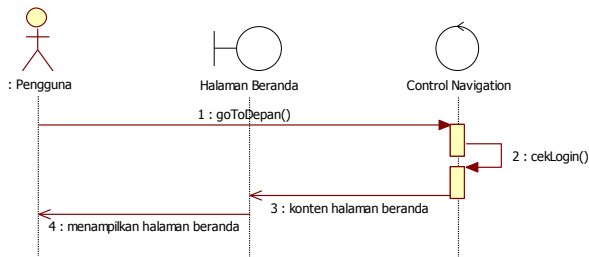
Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan diagram kelas, perancangan proses analisis, dan perancangan antarmuka.

### 3.2.1. Perancangan Diagram Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Lapisan pada arsitektur ini terdiri atas lapisan antarmuka, kontrol, dan data. Lapisan kontrol merupakan penghubung antara lapisan antarmuka dengan lapisan data. Subbab ini dibagi menjadi tiga bagian, yaitu diagram kelas untuk lapisan antarmuka, kontrol, dan data.

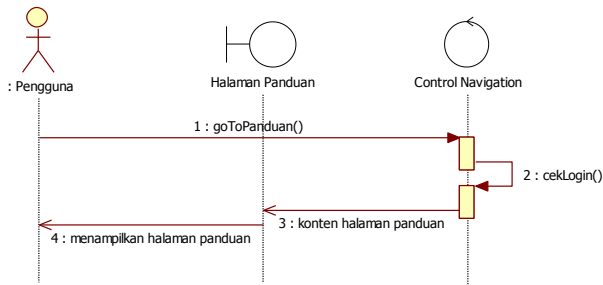
### 3.2.1.1. Diagram Kelas Lapisan Antarmuka

Karena aplikasi tugas akhir ini merupakan aplikasi web, oleh karena itu komponen dari lapisan antarmuka tidak dapat dideskripsikan dalam bentuk diagram kelas karena tidak ada kelas-kelas yang menyusun lapisan antarmuka. Oleh karena itu, untuk mendeskripsikan lapisan antarmuka dari aplikasi digunakan diagram urutan seperti misalnya proses pengguna membuka halaman beranda ditunjukkan oleh diagram urutan pada Gambar 3.10. Interaksi terjadi antara pengguna, halaman beranda, dan controller Navigation. Ketika pengguna melakukan klik link halaman beranda di halaman yang sedang diakses, maka fungsi `goToDepan` pada control Navigation akan dipanggil. Controller ini akan memanggil fungsi `cekLogin` yang akan mengecek status login dari pengguna dan mengarahkan ke halaman yang sesuai. Jika pengguna sudah login, maka pengguna akan diarahkan menuju halaman beranda anggota. Jika belum maka pengguna diarahkan menuju halaman beranda umum. Tampilan halaman dikirimkan controller ke halaman beranda sehingga dapat dilihat pengguna.

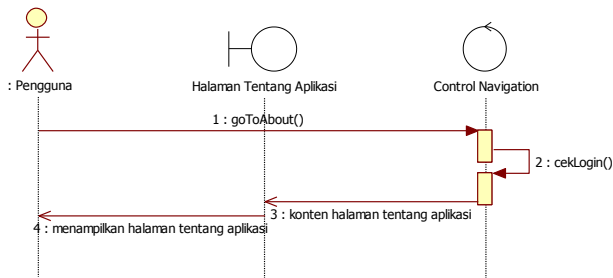


**Gambar 3.10 Diagram Urutan Proses Membuka Halaman Beranda**

Proses serupa juga berlaku pada proses membuka halaman panduan dan halaman tentang aplikasi seperti ditunjukkan berturut-turut pada Gambar 3.11 dan Gambar 3.12.

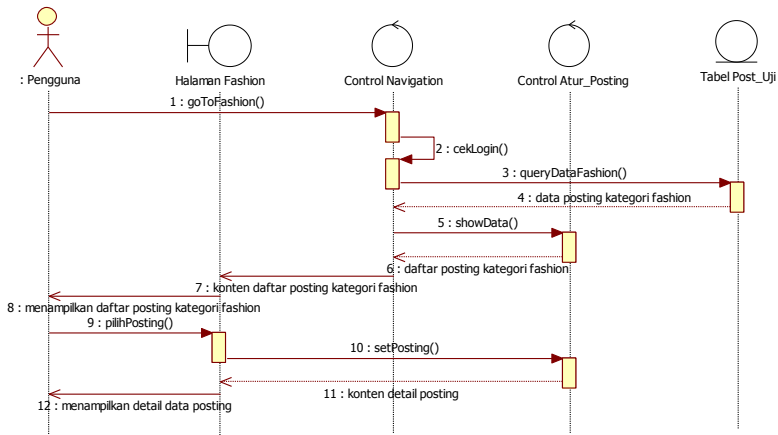


**Gambar 3.11 Diagram Urutan Proses Membuka Halaman Panduan**



**Gambar 3.12 Diagram Urutan Proses Membuka Halaman Tentang Aplikasi**

Proses navigasi ke halaman kategori barang, contohnya kategori fashion ditunjukkan oleh diagram urutan pada Gambar 3.13. Ketika pengguna melakukan klik link halaman fashion pada halaman yang sedang diakses, maka fungsi `goToFashion` pada control Navigation akan dipanggil. Sebelum melakukan perintah query ke tabel `Post_Uji`, akan dilakukan pengecekan status login dengan pemanggilan fungsi `cekLogin`. Setelah dilakukan query, tabel akan mengembalikan data posting kategori fashion. Data dalam bentuk objek ini akan diolah di control `Atur_Posting` oleh fungsi `showData` sehingga dapat ditampilkan dalam bentuk daftar posting. Hasil pengolahan tersebut akan dikirimkan kembali ke control Navigation untuk kemudian dikirimkan ke halaman fashion sehingga dapat dilihat oleh pengguna.

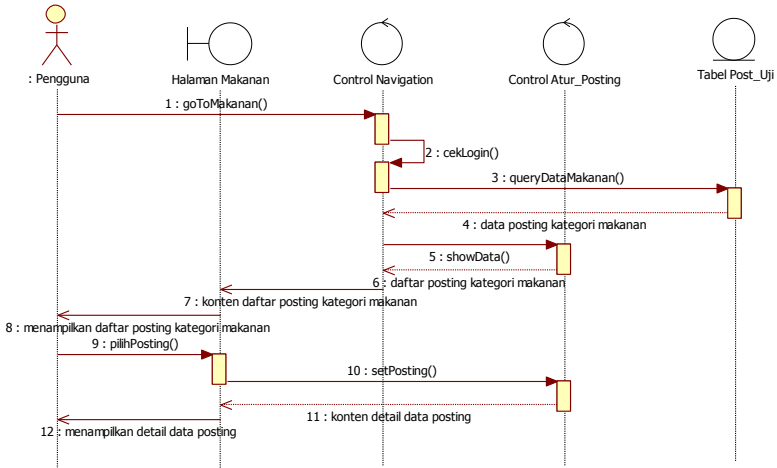


**Gambar 3.13 Diagram Urutan Membuka Halaman Kategori Fashion**

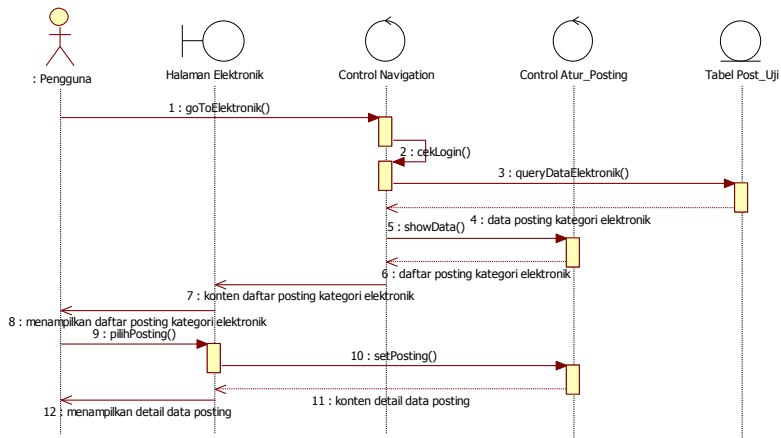
Apabila pengguna ingin melihat detail dari posting, pengguna dapat memilih salah satu posting dan kemudian fungsi `setPosting` di `control Atur_Posting` akan dipanggil. Fungsi ini akan mengembalikan nilai berupa konten dari detail posting yang dipilih pengguna. Konten tersebut akan dikirim ke halaman fashion sehingga dapat dilihat oleh pengguna. Proses serupa juga berlaku pada proses membuka halaman kategori makanan, kategori elektronik, kategori kendaraan, kategori jasa, dan kategori lain-lain seperti ditunjukkan berturut-turut pada Gambar 3.14, Gambar 3.15, Gambar 3.16, Gambar 3.17, dan Gambar 3.18.

Urutan proses melakukan pencarian barang pada aplikasi ditunjukkan pada Gambar 3.19. Setelah pengguna melakukan klik link halaman Cari Barang, maka pengguna akan diarahkan menuju halaman Cari Barang. Untuk melakukan pencarian barang, pengguna dapat mengisi form pencarian barang dan kemudian memilih tombol 'cari'. Ketika tombol 'cari' ditekan, maka fungsi `cariPost` di `control Cari` akan dipanggil. Fungsi tersebut akan melakukan query pada tabel `Post_Uji` yang akan mengembalikan data posting yang bersesuaian dengan kata kunci pengguna. Sebelum dikirim ke halaman Cari Barang, data tersebut diolah pada fungsi `showData` di

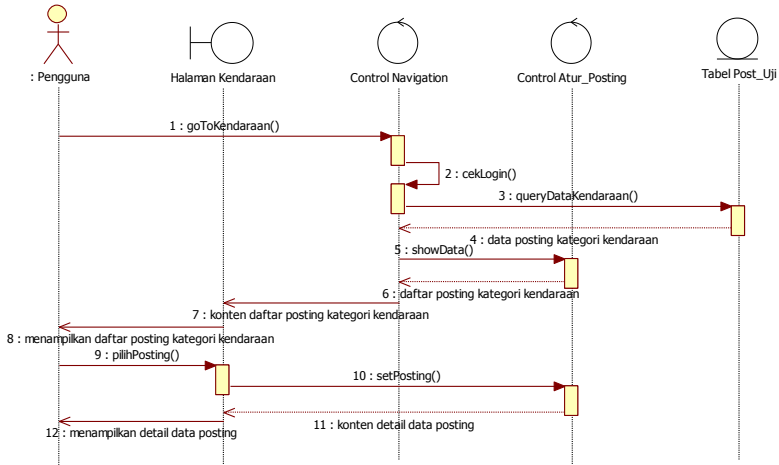
control Atur\_Posting sehingga siap ditampilkan pada panel hasil pencarian di halaman Cari Barang.



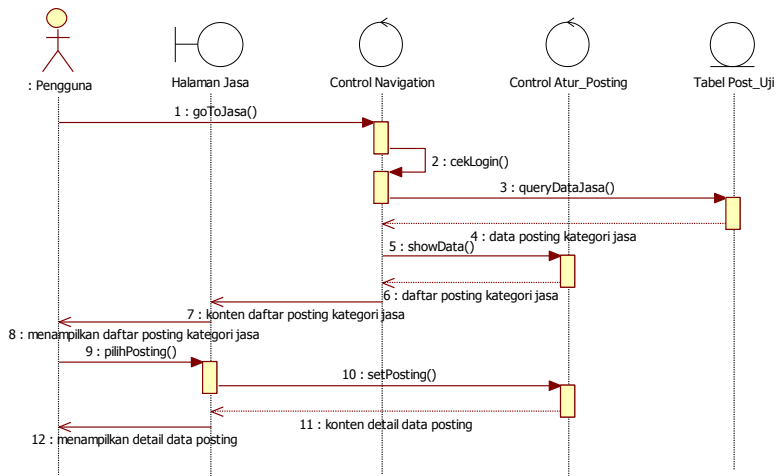
**Gambar 3.14 Diagram Urutan Membuka Halaman Kategori Makanan**



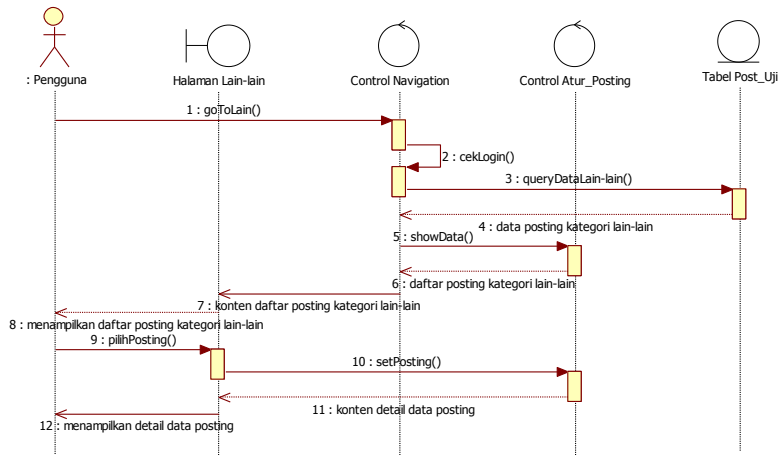
**Gambar 3.15 Diagram Urutan Membuka Halaman Kategori Elektronik**



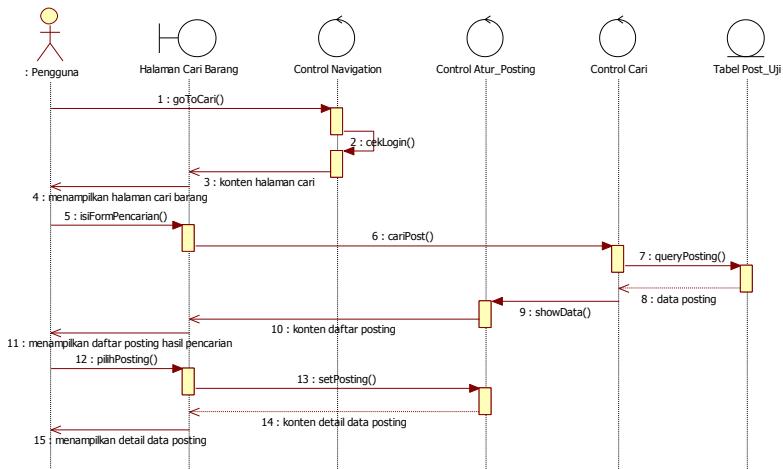
**Gambar 3.16 Diagram Urutan Membuka Halaman Kategori Kendaraan**



**Gambar 3.17 Diagram Urutan Membuka Halaman Kategori Jasa**



**Gambar 3.18 Diagram Urutan Membuka Halaman Kategori Lain-lain**

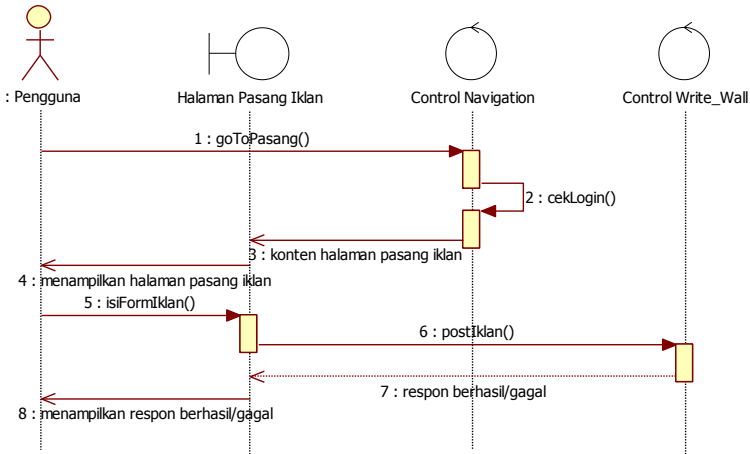


**Gambar 3.19 Diagram Urutan Proses Melakukan Pencarian Barang**

Apabila pengguna ingin melihat detail posting hasil pencarian, pengguna dapat memilih salah satu posting sehingga fungsi `setPosting` akan dipanggil dan mengembalikan nilai berupa detail dari posting yang diinginkan pengguna ke halaman Cari



Barang. Proses pengguna memasang iklan di wall grup melalui aplikasi ditunjukkan pada Gambar 3.20.

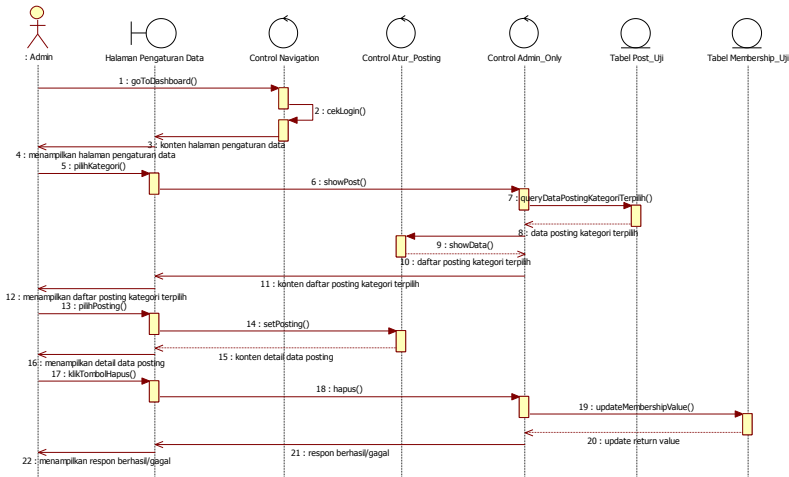


**Gambar 3.20 Diagram Urutan Proses Memasang Iklan Melalui Aplikasi**

Sebelum mengisi konten iklan, tentunya pengguna harus membuka halaman Pasang Iklan terlebih dahulu dengan cara melakukan klik link halaman Pasang Iklan. Setelah halaman Pasang Iklan ditampilkan pada pengguna, pengguna dapat mengisi form yang tersedia dimana pengguna dapat memasukkan konten iklan yang diinginkan. Ketika tombol ‘kirim’ pada form ditekan, maka fungsi `postIklan` pada `control Write_Wall` akan dipanggil. Fungsi ini akan mengirimkan konten iklan ke server Facebook dan akan memberikan nilai kembali berupa respon berhasil atau gagal dari Facebook. Respon tersebut akan dikirim ke halaman Pasang Iklan sehingga dapat dilihat oleh pengguna.

Proses admin aplikasi menghapus posting dari kategori pada menu Pengaturan Data ditunjukkan diagram urutan pada Gambar 3.21. Setelah admin membuka halaman menu Pengaturan Data, maka admin dapat memilih kategori yang ingin diolah datanya. Setelah memilih kategori, maka fungsi `showPost` pada `control Admin_Only` akan dipanggil dimana fungsi ini akan melakukan query pada tabel

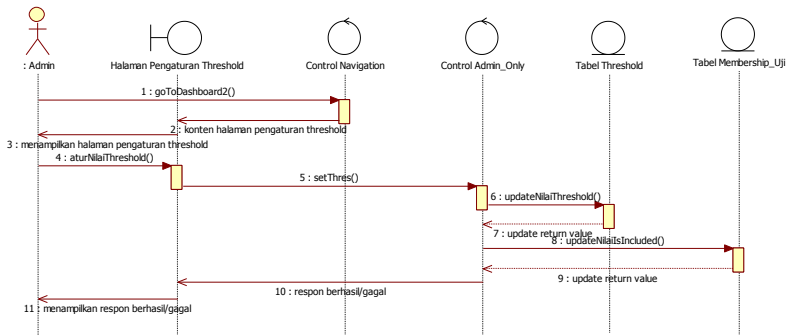
Post\_Uji untuk mendapatkan posting yang termasuk dalam kategori yang dipilih oleh admin. Hasil query tersebut akan dikembalikan ke control Admin\_Only dan kemudian dikirim ke fungsi showData pada control Atur\_Posting untuk diolah sehingga siap dikirim dan ditampilkan ke halaman Pengaturan Data.



**Gambar 3.21 Diagram Urutan Proses Menghapus Posting dari Kategori**

Untuk menghapus posting tertentu, terlebih dahulu admin harus memilih posting yang ingin dihapus. Setelah memilih posting, maka tampilan detail data posting akan ditampilkan. Untuk menghapus posting dari kategori, admin dapat memilih tombol 'hapus' pada halaman Pengaturan Data yang akan memanggil fungsi hapus pada control Admin\_Only. Fungsi ini akan meng-update membership value atau nilai keanggotaan dari posting yang dipilih pada kategori menjadi 0 sehingga posting tersebut secara langsung tidak akan muncul pada kategori. Proses ini akan memberikan nilai kembali berupa respon berhasil atau gagal yang kemudian akan dikirim ke halaman Pengaturan Data untuk ditampilkan pada pengguna.

Proses mengatur nilai threshold Fuzzy oleh admin ditunjukkan diagram urutan pada Gambar 3.22. Setelah admin membuka halaman Dashboard menu Pengaturan Threshold, admin dapat memilih mode pengaturan nilai yang diinginkan. Setelah selesai, admin dapat memilih tombol ‘set’ yang akan memanggil fungsi setThres pada control Admin\_Only. Fungsi ini akan meng-update nilai threshold Fuzzy pada tabel Threshold dan akan meng-update nilai is\_included yaitu flag yang menyatakan bahwa nilai keanggotaan dari posting di atas threshold yang ditentukan. Hanya posting dengan nilai keanggotaan di atas threshold yang akan ditampilkan oleh sistem. Setelah proses update selesai maka fungsi akan mengirim respon berhasil atau gagal dari proses ke halaman Pengaturan Threshold untuk disampaikan pada admin.

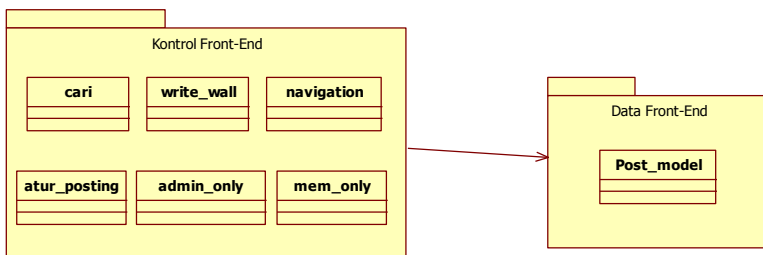


**Gambar 3.22 Diagram Urutan Proses Mengatur Nilai Threshold Fuzzy**

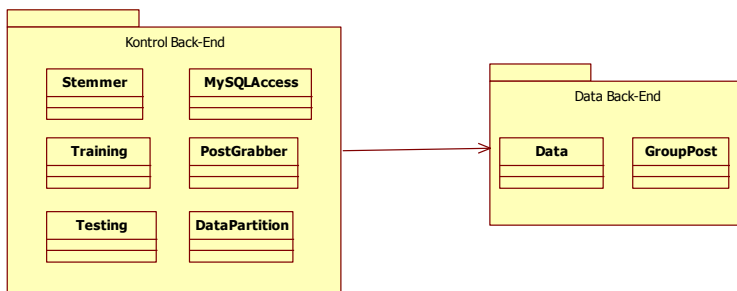
### 3.2.1.2. Diagram Kelas Lapisan Kontrol

Secara keseluruhan, hubungan antara package-package di dalam sistem dapat digambarkan dengan menggunakan package diagram. Sistem ini terbagi menjadi dua bagian yaitu front-end dan back-end. Package diagram front-end sistem ditunjukkan pada Gambar 3.23, package diagram back-end sistem ditunjukkan pada Gambar 3.24. Kelas-kelas pada package Kontrol Back-End akan berhubungan dengan kelas pada package Data Back-End. Kelas-kelas pada package Kontrol Front-End akan berhubungan dengan kelas pada package Data Front-End.

Dalam aplikasi ini terdapat dua jenis lapisan kontrol yaitu kontrol pada back-end aplikasi dan kontrol pada front-end aplikasi. Kontrol pada back-end aplikasi berfungsi melakukan pengolahan data posting yang kemudian akan ditampilkan pada pengguna melalui front-end. Diagram kelas lapisan kontrol back-end ditunjukkan pada Gambar 3.25. Sedangkan kontrol pada front-end aplikasi berfungsi sebagai jembatan antara user interface dengan data posting yang terdapat pada basis data. Diagram kelas lapisan kontrol front-end ditunjukkan pada Gambar 3.26.



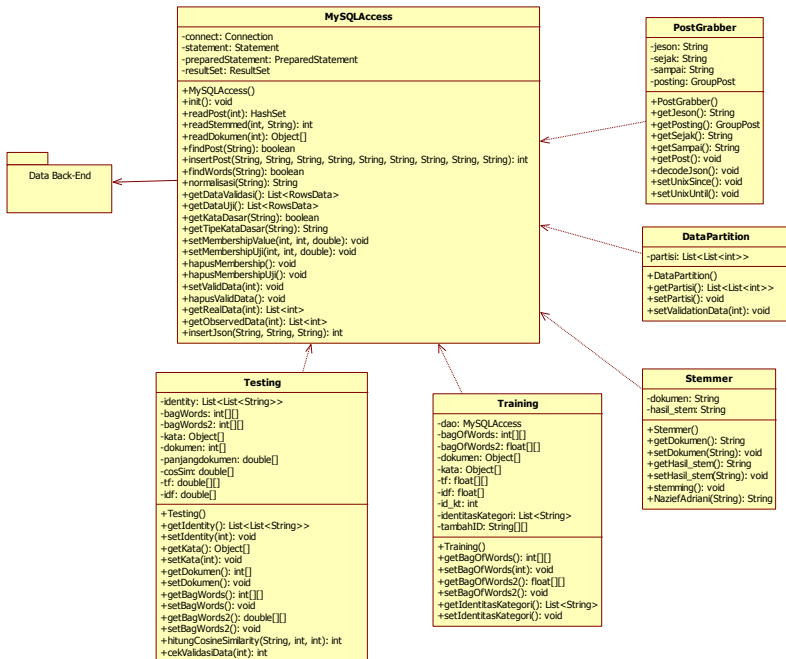
**Gambar 3.23 Package Diagram Front-End Sistem**



**Gambar 3.24 Package Diagram Back-End Sistem**

Kelas `MySQLAccess` pada kontrol back-end menangani semua proses yang berhubungan dengan koneksi data di basis data. Oleh karena itu kelas-kelas yang lain memiliki hubungan dependency dengan kelas ini karena kelas `MySQLAccess` menangani aliran data dari dan ke basis data sehingga ketika method di kelas lain memerlukan koneksi dengan basis data maka kelas `MySQLAccess`

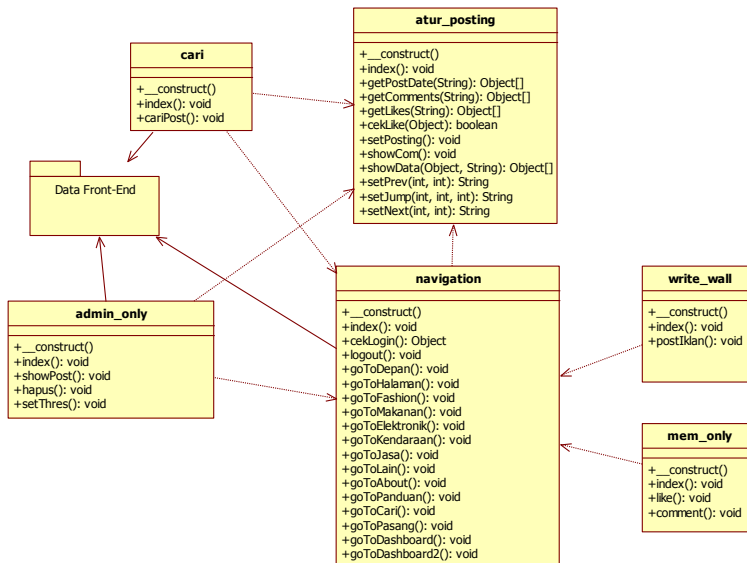
akan dipanggil dalam kelas tersebut. Kelas lain yang menyusun lapisan kontrol back-end antara lain kelas PostGrabber yang berfungsi mengambil data posting dari server Facebook, kelas Stemmer yang berfungsi melakukan proses stemming pada data posting, kelas DataPartition yang berfungsi melakukan partisi data yang akan digunakan dalam pembentukan model utama dengan menggunakan metode klasifikasi K-Fold Cross Validation, kelas Training yang digunakan untuk melakukan proses training data, serta kelas Testing yang digunakan untuk melakukan proses testing data pada proses klasifikasi.



**Gambar 3.25 Diagram Kelas Lapisan Kontrol Back-end**

Pada kontrol front-end terdapat kelas navigation yang merupakan kelas utama pada lapisan kontrol front-end. Kelas ini

menangani navigasi halaman-halaman web dari aplikasi. Semua perpindahan halaman pada aplikasi diatur pada kelas ini. Di dalam kelas ini juga terdapat fungsi cekLogin yang menangani proses pengecekan status login dari pengguna. Pengecekan status login diperlukan pada setiap proses di kelas lain. Oleh karena itu, kelas-kelas pada kontrol front-end yang lain memiliki hubungan dependency terhadap kelas navigation.



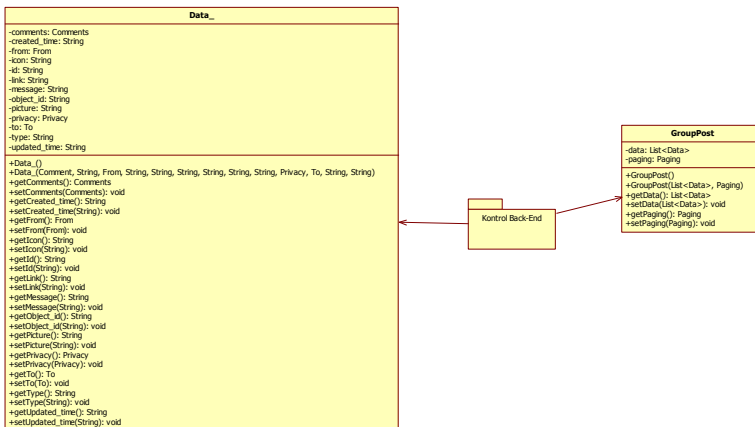
**Gambar 3.26 Diagram Kelas Lapisan Kontrol Front-end**

Selain itu terdapat kelas cari yang berfungsi melakukan pencarian posting tertentu berdasarkan kata kunci dari pengguna, kelas write\_wall yang menangani proses mem-posting iklan di grup, kelas mem\_only yang menangani proses yang hanya bisa dilakukan oleh anggota grup pada aplikasi yaitu memberikan like dan komentar pada posting, serta kelas admin\_only yang menangani proses yang hanya bisa dilakukan admin yaitu menghapus posting dari kategori dan mengatur nilai threshold Fuzzy. Selain itu juga terdapat kelas atur\_posting yang menangani proses pengaturan konten data posting

yang akan dikirimkan ke view melalui controller. Fungsi-fungsi yang terdapat pada kelas ini dipanggil oleh kelas lain. Oleh karena itu sejumlah kelas memiliki hubungan dependency dengan kelas `atur_posting` karena kelas tersebut menangani pengaturan tampilan data posting untuk dikirimkan dan ditampilkan oleh view aplikasi.

### 3.2.1.3. Diagram Kelas Lapisan Data

Dalam aplikasi ini terdapat dua jenis lapisan data yaitu lapisan data back-end aplikasi dan lapisan data front-end aplikasi. Lapisan data back-end aplikasi berisi kelas-kelas yang merepresentasikan struktur data posting yang didapatkan dari server Facebook. Terdapat dua kelas yaitu kelas `Data` dan kelas `GroupPost`. Kelas `Data` berisi data komponen-komponen dari posting seperti ID post, pesan, tanggal posting dibuat, tanggal posting di-update, dan lain-lain. Sedangkan kelas `GroupPost` berisi himpunan dari objek kelas `Data` yang diatur berdasarkan paging atau penomoran halaman. Gambar 3.27 menunjukkan diagram kelas lapisan data back-end.



**Gambar 3.27 Diagram Kelas Lapisan Data Back-end**

Sedangkan lapisan data front-end aplikasi berisi kelas yang merepresentasikan data posting yang telah diolah pada back-end

aplikasi. Terdapat satu kelas yaitu kelas `Post_model` yang berisi data hasil pengolahan posting yang akan ditampilkan di user interface. Gambar 3.28 menunjukkan diagram kelas lapisan data front-end.



**Gambar 3.28 Diagram Kelas Lapisan Data Front-end**



Pemetaan antara kelas-kelas yang terlibat di dalam sistem dengan fungsionalitas sistem yang mengimplementasikan kelas tersebut dijelaskan pada traceability matrix pada Tabel 3.10.

**Tabel 3.10 Traceability Matrix Sistem**

<b>Nama Kelas</b>	<b>Fungsionalitas Sistem</b>
<b>MySQLAccess</b>	Mengelompokkan posting berdasarkan kategori
<b>PostGrabber</b>	Mengelompokkan posting berdasarkan kategori
<b>Stemmer</b>	Mengelompokkan posting berdasarkan kategori
<b>DataPartition</b>	Mengelompokkan posting berdasarkan kategori
<b>Training</b>	Mengelompokkan posting berdasarkan kategori
<b>Testing</b>	Mengelompokkan posting berdasarkan kategori
<b>Navigation</b>	Melakukan login dengan akun Facebook
	Mengelompokkan posting berdasarkan kategori
<b>Cari</b>	Mencari posting tertentu
<b>Write_wall</b>	Mem-posting iklan
<b>Mem_only</b>	Memberi komentar pada posting
	Memberi like pada posting
<b>Admin_only</b>	Menghapus posting dari kategori
	Mengatur nilai threshold Fuzzy
<b>Atur_posting</b>	Mengelompokkan posting berdasarkan kategori
	Mencari posting tertentu
	Menghapus posting dari kategori

Pemetaan antara kelas-kelas yang terlibat di dalam sistem dengan halaman web aplikasi yang mengimplementasikan kelas tersebut dijelaskan pada Tabel 3.11.

**Tabel 3.11 Pemetaan Kelas dengan Halaman Web Aplikasi**

<b>Nama Kelas</b>	<b>Halaman Web Aplikasi</b>
<b>Navigation</b>	Halaman Beranda
	Halaman Panduan
	Halaman Tentang Aplikasi
	Halaman kategori fashion

	Halaman kategori makanan
	Halaman kategori elektronik
	Halaman kategori kendaraan
	Halaman kategori jasa
	Halaman kategori lain-lain
	Halaman Cari Barang
	Halaman Pasang Iklan
	Halaman Pengaturan Data
	Halaman Pengaturan Threshold
<b>Cari</b>	Halaman Cari Barang
<b>Write_wall</b>	Halaman Pasang Iklan
<b>Mem_only</b>	Halaman kategori fashion anggota
	Halaman kategori makanan anggota
	Halaman kategori elektronik anggota
	Halaman kategori kendaraan anggota
	Halaman kategori jasa anggota
	Halaman kategori lain-lain anggota
<b>Admin_only</b>	Halaman Pengaturan Data
	Halaman Pengaturan Threshold
<b>Atur_posting</b>	Halaman kategori fashion
	Halaman kategori makanan
	Halaman kategori elektronik
	Halaman kategori kendaraan
	Halaman kategori jasa
	Halaman kategori lain-lain
	Halaman Cari Barang
	Halaman Pasang Iklan
Halaman Pengaturan Data	

### 3.2.2. Perancangan Proses Analisis

Dalam subbab perancangan proses analisis akan dijelaskan tentang proses-proses yang berjalan pada aplikasi yang dibuat

berdasarkan kasus penggunaan. Perancangan proses dibuat dalam bentuk diagram alir.

### 3.2.2.1. Perancangan Proses Pengelompokan Data Posting

Proses pertama dalam pengolahan data posting adalah melakukan proses grab data posting dari server Facebook. Proses grab posting ini dijelaskan pseudocode pada Gambar 3.29. Diagram alir dari proses pengumpulan data posting ditunjukkan pada Gambar 3.31.

1. Melakukan HTTP *request* dengan *graph* API Facebook untuk mendapatkan data *posting*.
2. Data *posting* diterima dalam format JSON.
3. Melakukan proses *decode* JSON untuk mendapatkan data *posting* dalam bentuk objek.

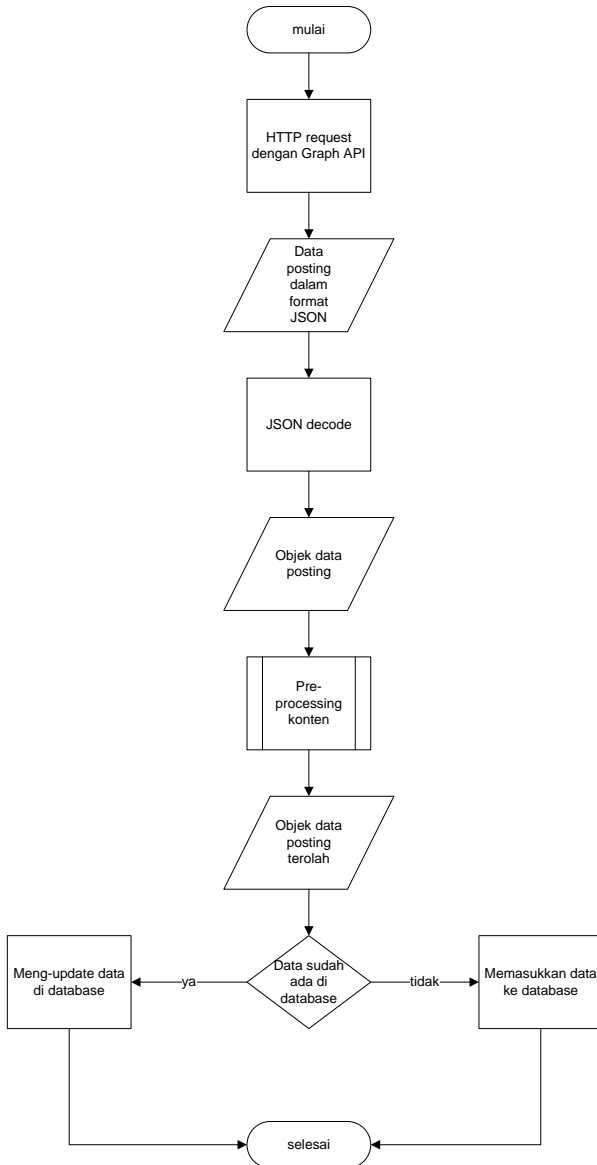
**Gambar 3.29 Pseudocode Proses Grab Posting**

Sebelum disimpan di basis data, dilakukan pra-proses dari konten posting tersebut. Pra-proses konten ini dijelaskan pseudocode pada Gambar 3.30. Diagram alir proses pra-proses konten ditunjukkan pada Gambar 3.32.

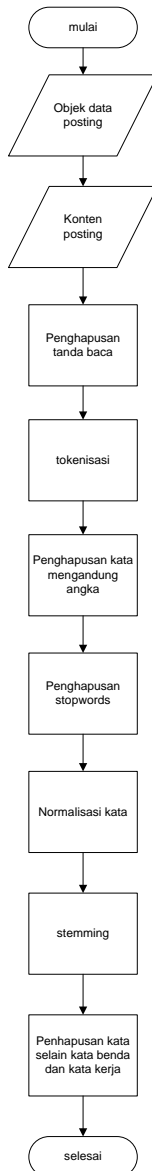
1. Penghapusan tanda baca dari konten.
2. Pemisahan kata-kata berdasarkan *delimiter* spasi/tokenisasi.
3. Penghapusan kata yang mengandung angka dari konten.
4. Penghapusan *stopwords* dari konten.
5. Proses normalisasi kata dengan melakukan pencocokan pada kata di *basis data*.
6. Proses *stemming* dengan algoritma Nazief Adriani.
7. Penghapusan kata yang bukan merupakan kata kerja atau kata benda.

**Gambar 3.30 Pseudocode Pra-proses Konten**

Setelah dilakukan pra-proses konten, data posting dimasukkan ke basis data dengan terlebih dahulu mengecek apakah posting tersebut sudah ada di basis data atau belum (menggunakan



**Gambar 3.31 Diagram Alir Proses Pengumpulan Data Posting**



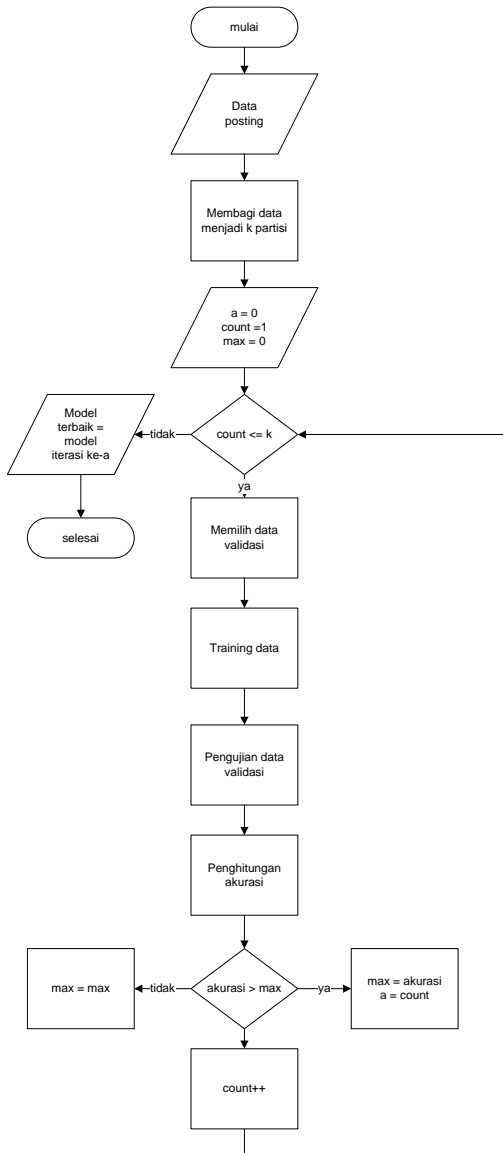
**Gambar 3.32 Diagram Alir Pra-proses Konten**

ID posting). Setelah didapatkan sejumlah data posting, tahap selanjutnya adalah membuat model dari data yang akan digunakan untuk melakukan pelabelan data uji. Model ini berupa kata-kata yang menjadi ciri spesifik setiap kategori yang merupakan label kelas data. Proses pembuatan model ini dilakukan dengan menggunakan metode K-Fold Cross Validation. Proses ini dijelaskan dengan pseudocode pada Gambar 3.33. Diagram alir metode K-Fold Cross Validation dapat dilihat pada Gambar 3.34. Pada setiap iterasi akan dihasilkan model yang dapat digunakan untuk pelabelan data uji. Dari sejumlah model hasil iterasi pada metode K-Fold Cross Validation tersebut, dipilih model dengan nilai akurasi tertinggi yang akan digunakan sebagai model untuk pelabelan data uji.

1. Membagi data *posting* menjadi  $k$  partisi
2. For  $\leftarrow k$ 
  - Memilih sebuah partisi yang akan dijadikan data validasi
  - Melakukan *training* data pada partisi selain data validasi
  - Data validasi diuji dengan model yang didapat dari hasil *training* data partisi yang lain
  - Menghitung nilai akurasi dari hasil yang didapat
3. Memilih model dengan akurasi terbaik

**Gambar 3.33 Pseudocode Metode K-Fold Cross Validation**

Setelah didapatkan model dari setiap kategori, maka model siap digunakan untuk proses pelabelan kategori untuk data uji selanjutnya. Proses pelabelan kelas dari data uji dilakukan dengan mengukur tingkat kedekatan antara model dengan data uji. Pengukuran jarak antara model dengan data uji ini dilakukan dengan menggunakan metode perhitungan Cosine Similarity yang dijelaskan pada pseudocode pada Gambar 3.35. Jarak antara model dari setiap kategori dengan data uji tersebut merupakan nilai keanggotaan dari data terhadap kategori terkait yang menunjukkan tingkat relevansi dari data terhadap kategori tersebut.



**Gambar 3.34 Diagram Alir Metode K-Fold Cross Validation**

1.  $A$  = perkalian *dot product* dari vektor model dan vektor data uji.
2.  $L\_model$  = akar dari penjumlahan kuadrat vektor model.
3.  $L\_data$  = akar dari penjumlahan kuadrat vektor data uji.
4.  $cosSim = A / (L\_model \times L\_data)$ .

**Gambar 3.35 Pseudocode Perhitungan Cosine Similarity**

### 3.2.2.2. Perancangan Proses Login dengan Akun Facebook

Proses login dengan akun Facebook dimulai dengan pengecekan apakah pengguna sudah login atau belum. Apabila pengguna belum login dengan akun Facebook, maka sistem akan menampilkan tombol untuk melakukan login. Sedangkan jika pengguna sudah login, maka sistem akan menampilkan nama dan foto profil dari pengguna serta mengarahkan pengguna ke halaman anggota. Proses login ini dijelaskan pseudocode pada Gambar 3.36. Diagram alir dari proses login dengan akun Facebook ditunjukkan pada Gambar 3.37.

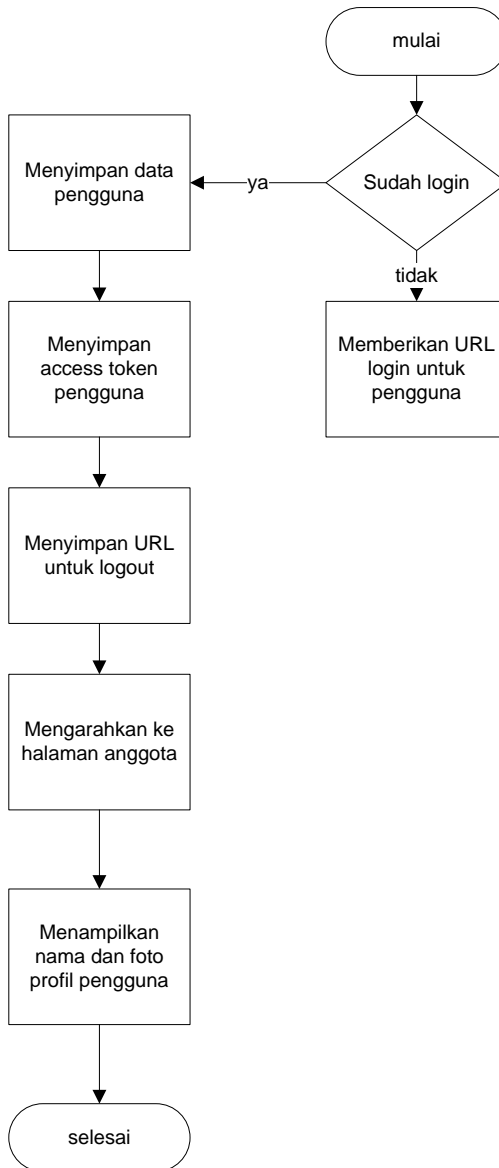
1. Melakukan pengecekan apakah pengguna sudah *login* dengan akun Facebook.
2. *If* sudah *login*
  - Menyimpan data pengguna.
  - Menyimpan *access token* dari pengguna.
  - Menyimpan URL untuk *logout* bagi pengguna.
  - Mengarahkan pengguna ke halaman anggota.
  - Menampilkan nama dan foto profil dari pengguna.
3. *Else*
  - Memberikan URL *login* untuk pengguna.

**Gambar 3.36 Pseudocode Proses Login dengan Akun Facebook**

### 3.2.2.3. Perancangan Proses Mem-posting Iklan

Untuk dapat mem-posting iklan, pengguna harus login dengan akun Facebook terlebih dahulu. Mula-mula sistem akan melakukan pengecekan konten yang akan di-posting. Terdapat tiga





**Gambar 3.37 Diagram Alir Proses Login dengan Akun Facebook**

komponen yang dapat di-posting, yaitu teks, link, dan gambar. Untuk gambar yang akan di-posting diperoleh dari proses upload file oleh pengguna. Proses mem-posting iklan ini dijelaskan pseudocode pada Gambar 3.38. Diagram alir dari proses mem-posting iklan ditunjukkan pada Gambar 3.40.

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Melakukan pengecekan konten iklan yang akan di-<i>posting</i>.</li> <li>2. <i>If</i> mengandung gambar             <ul style="list-style-type: none"> <li>• Meng-<i>upload</i> gambar ke <i>server</i> aplikasi.</li> <li>• Mencatat lokasi gambar di <i>server</i> aplikasi.</li> <li>• Mengirimkan lokasi gambar beserta komponen konten yang lain ke <i>server</i> Facebook.</li> </ul> </li> <li>3. <i>Else</i> <ul style="list-style-type: none"> <li>• Mengirimkan konten iklan ke <i>server</i> Facebook.</li> </ul> </li> <li>4. Menampilkan <i>error log</i> dari Facebook jika ada.</li> </ol> |
|--|

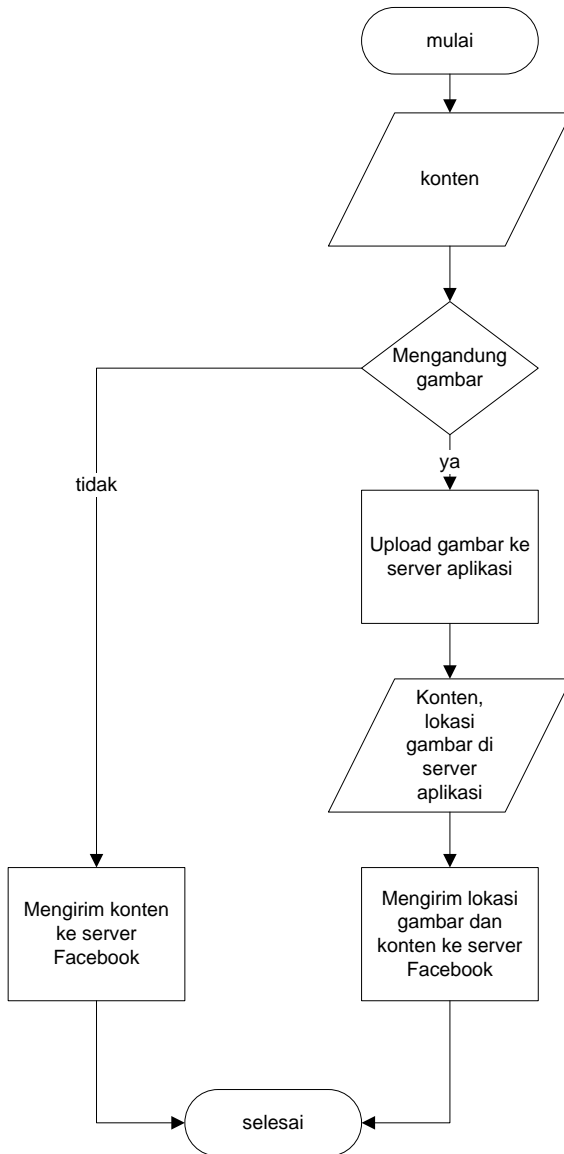
**Gambar 3.38 Pseudocode Proses Mem-posting Iklan**

### 3.2.2.3. Perancangan Proses Memberikan Komentar pada Posting

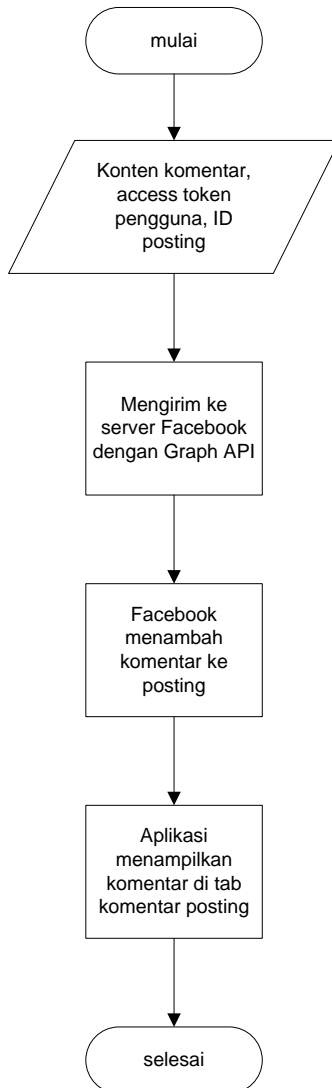
Untuk dapat memberikan komentar pada posting maka pengguna harus login dengan menggunakan akun Facebook terlebih dahulu. Setelah melakukan login, pengguna dapat memilih posting yang akan diberi komentar dan menulis komentar pada tab komentar yang tersedia. Proses memberikan komentar pada posting ini dijelaskan pada pseudocode pada Gambar 3.39. Diagram alir dari proses memberikan komentar ditunjukkan pada Gambar 3.41.

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Mencatat konten komentar yang ditulis pengguna.</li> <li>2. Mencatat <i>access token</i> dari pengguna.</li> <li>3. Mencatat ID <i>posting</i> yang diberi komentar.</li> <li>4. Mengirim konten komentar, <i>access token</i> pengguna, dan ID <i>posting</i> dengan <i>Graph API</i> ke <i>server</i> Facebook.</li> <li>5. Facebook menambahkan komentar ke <i>posting</i>.</li> <li>6. Menampilkan komentar pengguna ke <i>tab</i> komentar <i>posting</i>.</li> </ol> |
|--|

**Gambar 3.39 Pseudocode Proses Memberi Komentar pada Posting**



**Gambar 3.40 Diagram Alir Proses Mem-posting Iklan**



**Gambar 3.41 Diagram Alir Proses Memberi Komentar pada Posting**

### 3.2.2.4. Perancangan Proses Memberikan Like pada Posting

Setelah login dengan akun Facebook, pengguna dapat memilih posting yang ingin di-like kemudian menekan tombol 'Suka' yang tersedia. Proses memberikan like pada posting ini dijelaskan pada pseudocode pada Gambar 3.42. Diagram alir dari proses memberikan komentar ditunjukkan pada Gambar 3.44.

1. Mencatat *access token* dari pengguna.
2. Mencatat ID *posting* yang diberi *like*.
3. Mengirim *access token* pengguna dan ID *posting* dengan Graph API ke *server* Facebook.
4. Facebook menambahkan jumlah *like* pada *posting*.
5. Meng-*update* jumlah *like* pada *posting*.

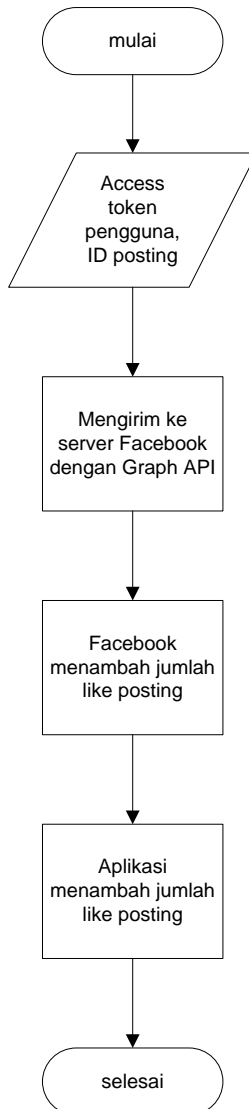
**Gambar 3.42 Pseudocode Proses Memberi Like pada Posting**

### 3.2.2.5. Perancangan Proses Menghapus Posting dari Kategori

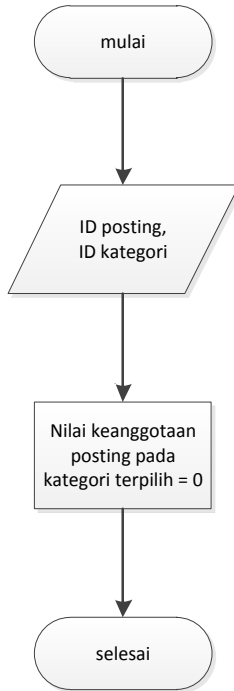
Proses ini hanya dapat dilakukan oleh admin aplikasi. Proses ini untuk mengatasi kelemahan proses klasifikasi yang tidak selalu menghasilkan keluaran yang sesuai dengan yang diharapkan dimana admin dapat menghapus posting yang tidak sesuai pengelompokannya dari kategori. Setelah login, maka admin dapat memilih posting yang ingin dihapus pada menu Pengaturan Data di Dashboard. Proses menghapus posting dari kategori dijelaskan pada pseudocode pada Gambar 3.43. Diagram alir dari proses menghapus posting dari kategori ditunjukkan pada Gambar 3.45.

1. Mencatat ID *posting* yang ingin dihapus
2. Mencatat ID kategori terpilih
3. Mengubah nilai keanggotaan dari *posting* dengan ID terpilih pada kategori terpilih menjadi 0

**Gambar 3.43 Pseudocode Proses Menghapus Posting dari Kategori**



**Gambar 3.44 Diagram Alir Proses Memberi Like pada Posting**



**Gambar 3.45 Diagram Alir Proses Menghapus Posting dari Kategori**

### **3.2.2.6. Perancangan Proses Mengatur Nilai Threshold Fuzzy**

Proses ini hanya dapat dilakukan oleh admin aplikasi. Setelah login, maka admin dapat memilih metode pengaturan nilai threshold pada menu Pengaturan Threshold di Dashboard. Terdapat empat metode yang tersedia yaitu memilih mengambil nilai keanggotaan tertinggi pada setiap data, mengambil dua nilai keanggotaan tertinggi pada setiap data, mengambil nilai rata-rata nilai keanggotaan pada setiap data, atau mengatur nilai threshold Fuzzy secara manual. Proses mengatur nilai threshold Fuzzy ini dijelaskan pada pseudocode pada Gambar 3.46. Diagram alir dari proses ditunjukkan pada Gambar 3.47.

```

1. Foreach data uji
  • If pilih metode 1
    • Beri flag pada data dengan nilai
      keanggotaan tertinggi.
  • Else if pilih metode 2
    • Beri flag pada data dengan dua nilai
      keanggotaan tertinggi.
  • Else if pilih metode 3
    • Hitung rata-rata nilai keanggotaan data
      pada tiap kategori.
    • Beri flag pada data dengan nilai
      keanggotaan di atas rata-rata.
  • Else
    • Mencatat nilai threshold yang diberikan
      pengguna.
    • Beri flag pada data dengan nilai
      keanggotaan di atas nilai threshold.
2. Tampilkan posting yang telah diberi flag pada
   aplikasi

```

**Gambar 3.46 Pseudocode Proses Mengatur Nilai Threshold Fuzzy**

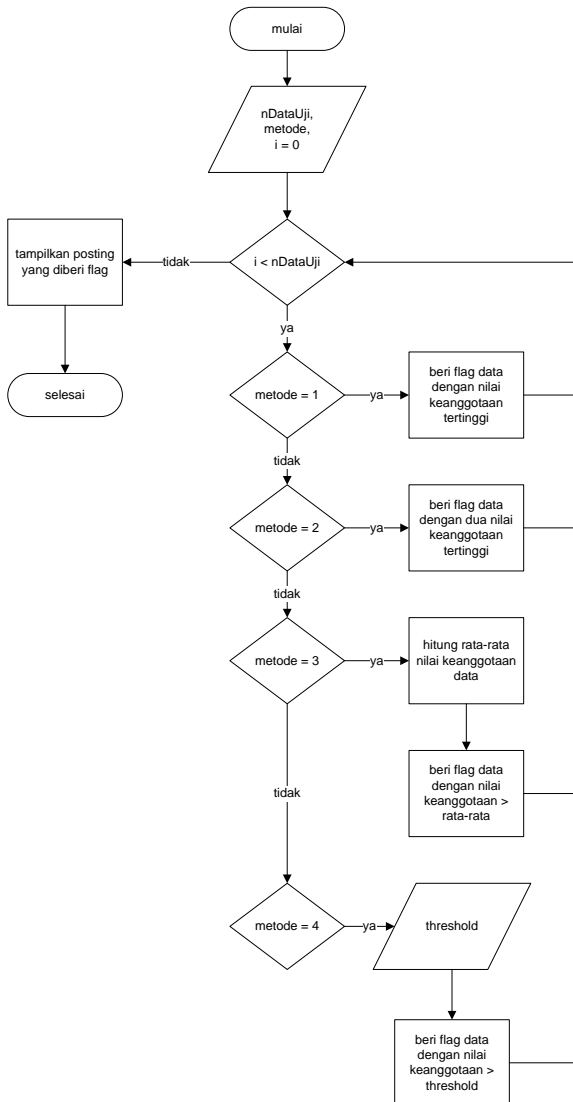
### 3.2.3. Perancangan Antarmuka Pengguna

Bagian ini membahas rancangan tampilan antarmuka pada sistem. Pada sistem ini terdapat lima jenis tampilan yang akan dijelaskan pada subbab berikut.

#### 3.2.3.1. Rancangan Antarmuka Halaman Beranda

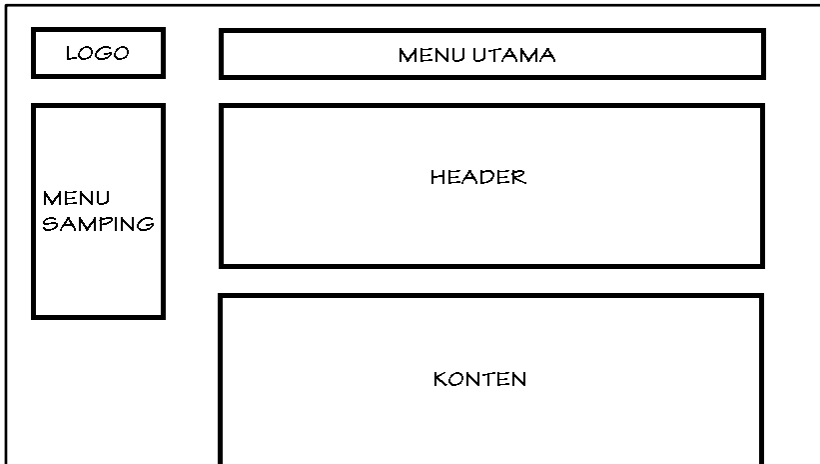
Halaman ini merupakan tampilan awal ketika pengguna membuka aplikasi. Pada halaman ini terdapat menu utama yang terletak pada bagian atas dari halaman yang dapat digunakan sebagai navigasi dari fitur-fitur pada aplikasi. Terdapat empat navigasi utama yaitu halaman Beranda, halaman Cari Barang, halaman Tentang Aplikasi, dan halaman Panduan. Apabila pengguna telah melakukan login dengan akun Facebook, maka akan terdapat navigasi tambahan yaitu menuju halaman Pasang Iklan. Selain itu pada bagian samping terdapat sub menu untuk menuju halaman data posting kategori sesuai kategori yang tersedia. Baik menu utama maupun menu samping ini tersedia pada semua halaman dari aplikasi.





**Gambar 3.47 Diagram Alir Proses Mengatur Nilai Threshold Fuzzy**

Tampilan rancangan antarmuka halaman beranda ditunjukkan pada Gambar 3.48.



**Gambar 3.48 Rancangan Antarmuka Halaman Beranda**

### **3.2.3.2. Rancangan Antarmuka Halaman Pencarian Barang**

Halaman ini merupakan antarmuka yang digunakan ketika pengguna hendak mencari posting berdasarkan kata kunci tertentu. Pada halaman ini terdapat form pencarian barang dimana terdapat textbox yang digunakan sebagai tempat memasukkan kata kunci dari pengguna serta sebuah tombol yang akan memberi perintah pada sistem untuk melakukan pencarian. Hasil pencarian akan ditampilkan pada panel di bawah form pencarian. Tampilan rancangan antarmuka halaman pencarian barang ditunjukkan pada Gambar 3.49.

### **3.2.3.3. Rancangan Antarmuka Halaman Pemasangan Iklan**

Halaman ini hanya dapat diakses oleh pengguna yang login dengan akun Facebook. Pada halaman ini terdapat form penulisan iklan dimana terdapat sebuah textarea untuk menulis konten iklan, sebuah file uploader yang digunakan untuk meng-upload gambar

pengguna, dan sebuah textbox untuk menulis link yang ingin disertakan pada iklan. Selain itu juga terdapat tombol yang memberi perintah pengiriman konten iklan ke server Facebook. Tampilan rancangan antarmuka halaman pemasangan iklan dapat dilihat pada Gambar 3.50.



The diagram shows a web interface layout for a search page. It consists of a main container with a left sidebar and a main content area. The sidebar contains a 'LOGO' box at the top and a 'MENU SAMPING' box below it. The main content area is divided into three horizontal sections: 'MENU UTAMA' at the top, 'FORM PENCARIAN BARANG' in the middle, and 'PANEL HASIL PENCARIAN' at the bottom. The search form includes a text input field and a 'cari' button.

**Gambar 3.49 Rancangan Antarmuka Halaman Pencarian Barang**

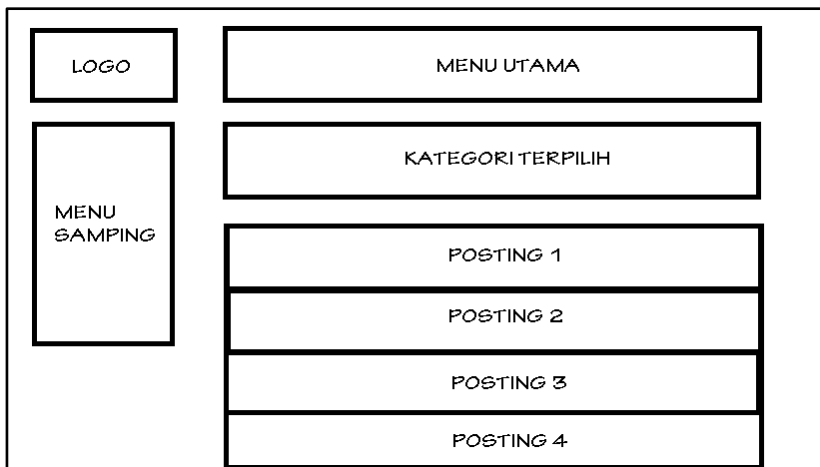


The diagram shows a web interface layout for an ad placement page. It features a sidebar on the left with 'LOGO' and 'MENU SAMPING' boxes. The main content area has a 'MENU UTAMA' box at the top and a 'FORM PENULISAN IKLAN' section below. This form contains three labeled input fields: 'konten' (a large text area), 'gambar' (with a 'browse' button), and 'link' (a text input field). A 'KIRIM' button is positioned at the bottom of the form.

**Gambar 3.50 Rancangan Antarmuka Halaman Pemasangan Iklan**

#### 3.2.3.4. Rancangan Antarmuka Halaman Data Posting

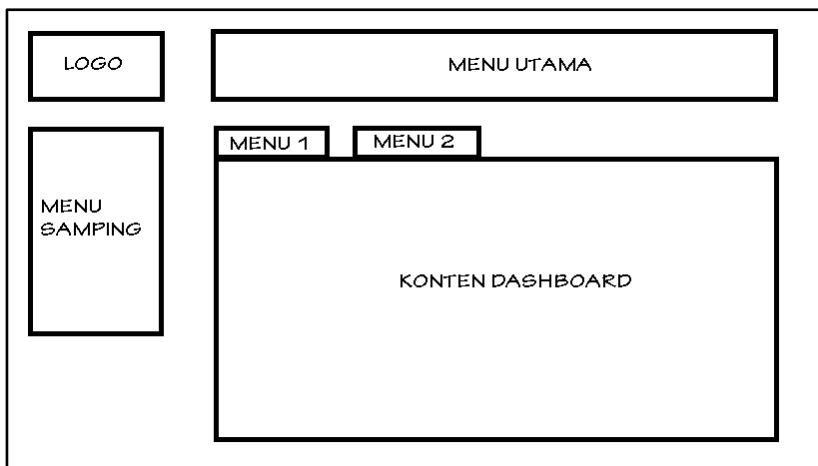
Halaman ini berisi data posting yang telah dikelompokkan berdasarkan kategori. Halaman ini akan menampilkan daftar data posting yang terkelompokkan berdasarkan kategori data terpilih. Apabila data posting tertentu dipilih maka pengguna akan diarahkan untuk melihat detail dari posting tersebut. Tampilan rancangan antarmuka halaman ini ditunjukkan pada Gambar 3.51.



Gambar 3.51 Rancangan Antarmuka Halaman Data Posting

#### 3.2.3.5. Rancangan Antarmuka Halaman Dashboard Admin

Halaman ini merupakan halaman khusus admin aplikasi dimana admin dapat melakukan sejumlah pengaturan di antaranya pengaturan nilai threshold Fuzzy serta pengaturan data posting. Untuk pengaturan nilai threshold Fuzzy, admin dapat memilih dari empat mode pengaturan yang tersedia. Untuk pengaturan data posting, admin dapat memilih untuk menghapus data posting yang tidak sesuai penempatan kategorinya. Tampilan rancangan antarmuka halaman Dashboard ditunjukkan pada Gambar 3.52.



**Gambar 3.52 Rancangan Antarmuka Halaman Dashboard Admin**

## **BAB IV IMPLEMENTASI**

Bab ini membahas tentang implementasi dari perancangan sistem. Bab ini berisi proses implementasi dari setiap kelas pada semua modul. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java dan PHP.

### **4.1. Implementasi Lapisan Antarmuka**

Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem. Untuk menyusun tampilan dari aplikasi ini digunakan bahasa markah HTML (HyperText Markup Language) yang dikombinasikan dengan CSS (Cascading Style Sheet) dan JavaScript untuk meningkatkan kesan interaktif dari aplikasi.

### **4.2. Implementasi Lapisan Kontrol**

Lapisan kontrol merupakan lapisan yang bertanggung jawab dengan tingkah laku sistem. Lapisan ini bertugas menghubungkan lapisan data dengan lapisan antarmuka.

#### **4.2.1. Kelas MySQLAccess**

Kelas ini merupakan kelas yang bertugas menangani proses yang berhubungan dengan basis data. Dalam aplikasi ini, basis data yang digunakan adalah basis data MySQL. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.1. Dalam kelas ini terdapat fungsi `init` yang merupakan tahap inisialisasi dari koneksi dengan basis data. Di antaranya adalah mendeklarasikan koneksi yang digunakan, basis data yang terhubung dengan sistem, serta username dan password yang digunakan untuk mengakses basis data. Selain itu terdapat sejumlah fungsi-fungsi lain seperti `readPost`, `readStemmed`, `readDokumen`, `findPost`, dan fungsi-fungsi yang lain yang berhubungan dengan pengolahan data dari atau ke basis data. Fungsi-fungsi di dalam kelas ini dipanggil di dalam kelas yang lain dalam kaitannya dengan pengolahan data.

#### **4.2.2. Kelas PostGrabber**

Kelas ini merupakan kelas yang bertugas menangani proses pengambilan data posting dari server Facebook. Di dalam kelas ini terdapat fungsi `getPost` yang melakukan HTTP request ke server Facebook untuk mendapatkan data posting dari grup. Request ini mengirimkan parameter berupa ID grup jual beli dan juga access token aplikasi. Facebook memberikan respon berupa data posting dalam format JSON. Selain itu pada kelas ini juga terdapat fungsi `decodeJson` yang melakukan proses decode string JSON dari Facebook ke dalam objek Data yaitu objek posting pada aplikasi. Dalam kelas ini juga terdapat fungsi `setUnixSince` dan `setUnixUntil` yang digunakan untuk mencatat paging data posting yang dikirimkan oleh Facebook. Rincian kelas ini dapat dilihat pada Kode Sumber 7.2.

#### **4.2.3. Kelas Stemmer**

Kelas `Stemmer` bertugas melakukan pra-proses konten data posting yang telah didapat dari server Facebook. Seperti nama kelasnya, proses utama yang dilakukan di kelas ini adalah proses stemming. Sebelum dilakukan stemming, terlebih dahulu dilakukan pengolahan seperti penghapusan tanda baca, pemisahan kata berdasarkan delimiter, penghapusan stopwords, dan proses normalisasi untuk kata-kata yang dianggap kurang baku. Setelah itu dilakukan proses stemming dengan menggunakan algoritma Nazief Adriani. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.3.

#### **4.2.4. Kelas DataPartition**

Kelas `DataPartition` bertugas melakukan partisi data posting yang telah diperoleh untuk kemudian digunakan pada proses klasifikasi dengan metode K-Fold Cross Validation. Data posting yang akan diolah dengan metode ini sejumlah 200 data. Data ini akan dibagi menjadi 10 partisi dan akan diolah dalam 10 iterasi dimana setiap partisi akan menjadi data uji sebanyak satu kali dan data training sebanyak 9 kali. Di dalam kelas ini terdapat

fungsi `setPartisi` yang akan melakukan pembagian data ke dalam 10 partisi dimana setiap kategori terwakili oleh sejumlah data. Selain itu juga terdapat fungsi `setValidationData` yang bertugas memberi flag pada setiap iterasi pada partisi yang akan digunakan sebagai data uji. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.4.

#### **4.2.5. Kelas Training**

Kelas ini merupakan kelas yang menangani proses training data. Proses training data dilakukan pada setiap kategori secara bergantian. Proses dimulai dengan pembentukan bag of words yang mendaftarkan jumlah kemunculan kata dalam setiap dokumen. Proses ini diimplementasikan oleh fungsi `setBagOfWords`. Dengan mendaftarkan jumlah kemunculan setiap kata, maka dapat dihitung bobot dari setiap kata dengan menggunakan perhitungan TF-IDF seperti yang dijelaskan pada subbab 2.5. Proses perhitungan TF-IDF diimplementasikan pada fungsi `setBagOfWords2`. Dari perhitungan bobot tiap kata tersebut akan dipilih sejumlah kata dengan bobot tertinggi yang akan digunakan sebagai identitas dari kategori yang nantinya akan digunakan sebagai model untuk melakukan pengujian data. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.5.

#### **4.2.6. Kelas Testing**

Kelas ini merupakan kelas yang menangani proses testing data. Proses testing data dimulai dengan pembentukan bag of words dari model yang didapat dari proses training data untuk mendaftarkan bobot dari tiap kata yang ada pada model, proses ini diimplementasikan pada fungsi `setBagWords` dan `setBagWords2`. Fungsi `setBagWords` digunakan untuk mendaftarkan jumlah kemunculan kata dalam model dan fungsi `setBagWords2` digunakan untuk perhitungan bobot dari kata pada model dengan menggunakan perhitungan TF-IDF. Kemudian untuk setiap data uji, dilakukan perhitungan jumlah kemunculan kata yang disesuaikan dengan model. Perhitungan ini kemudian akan digunakan untuk menghitung tingkat kedekatan data uji dengan model dengan menggunakan



perhitungan Cosine Similarity seperti yang dijelaskan pada subbab 2.8. Nilai yang menyatakan tingkat kedekatan data uji terhadap model dari tiap kategori tersebut yang akan digunakan sebagai nilai keanggotaan dari data uji tersebut terhadap kategori terkait. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.6. Kode sumber untuk fungsi perhitungan Cosine Similarity dari data uji dapat dilihat pada Kode Sumber 7.7.

#### **4.2.7. Kelas Navigation**

Kelas ini merupakan kelas controller yang menangani perpindahan halaman pada front-end aplikasi. Semua proses redirecting page diatur di dalam kelas ini. Di dalam kelas ini juga terdapat metode untuk mengecek apakah pengguna sudah login dengan akun Facebook atau belum yaitu pada fungsi cekLogin. Proses pengecekan ini berguna untuk mengarahkan pengguna ke halaman yang sesuai. Apabila pengguna sudah login, maka pengguna akan diarahkan menuju halaman anggota dan pengguna akan mendapatkan hak akses dari anggota seperti dapat mem-posting iklan, menulis komentar, atau memberikan like.

Proses pengaturan konten yang akan ditampilkan dalam halaman aplikasi juga diatur dalam kelas ini. Data diambil dari basis data dengan melakukan load model yang kemudian melakukan query data dan mengembalikan data posting yang kemudian akan dikirim oleh controller ke view. Kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 7.8.

#### **4.2.8. Kelas Cari**

Kelas cari merupakan kelas controller yang menangani proses pencarian posting berdasarkan kata kunci dari pengguna yang merupakan salah satu fitur dari aplikasi. Pada kelas ini terdapat fungsi cariPost yang melakukan load model yang kemudian melakukan query data untuk mencari posting berdasarkan parameter berupa kata kunci dari pengguna dan akan mengembalikan data posting yang bersesuaian untuk kemudian dikirim ke view untuk ditampilkan. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.9.

#### **4.2.9. Kelas Write\_wall**

Kelas ini merupakan kelas controller yang menangani proses menulis posting di wall grup melalui aplikasi. Fitur ini ditangani oleh fungsi postIklan dimana fungsi ini akan menerima input dari pengguna berupa konten iklan yang akan di-post. Terdapat tiga komponen iklan yang dapat ditulis oleh pengguna yaitu teks, gambar, dan link. Fungsi ini akan mengecek apakah konten yang dikirim pengguna mengandung gambar atau tidak. Bila konten mengandung gambar maka gambar yang di-upload oleh pengguna akan di-upload ke server aplikasi. Lokasi file ini yang kemudian akan dicatat dan dikirim ke server Facebook dengan menggunakan Graph API bersama dengan komponen konten yang lain. Apabila tidak mengandung gambar, maka konten akan langsung dikirim ke server Facebook. Fungsi ini akan memberikan nilai kembali berupa respon dari sistem baik berupa pesan error jika gagal atau pesan berhasil jika proses berhasil. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.10.

#### **4.2.10. Kelas Mem\_only**

Kelas ini menangani proses yang hanya bisa dilakukan oleh pengguna yang memiliki hak akses anggota. Proses seperti memberi like dan memberi komentar hanya bisa dilakukan oleh pengguna yang merupakan anggota dari grup Lapak Jual Beli ITS di Facebook dan melakukan login di aplikasi. Dalam kelas ini terdapat fungsi like yang menangani proses memberi like pada posting. Ketika tombol 'Suka' di halaman web ditekan maka fungsi ini akan dipanggil dan akan mengirimkan access token dari pengguna dan ID dari posting yang dipilih ke server Facebook dengan Graph API. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.11.

Selain itu terdapat fungsi comment yang menangani proses memberi komentar pada posting. Fungsi ini menerima input berupa konten komentar yang ditulis pengguna, ID posting, dan access token dari pengguna yang akan dikirim ke server Facebook dengan Graph API. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.12.

#### **4.2.11. Kelas Admin\_only**

Kelas ini menangani proses yang hanya bisa dilakukan oleh pengguna yang memiliki hak akses sebagai admin. Admin dapat menghapus posting dari data posting kategori tertentu apabila dianggap posting tersebut tidak sesuai berada di kategori tersebut. Fitur ini dapat menangani kelemahan proses klasifikasi yang tidak selalu menghasilkan keluaran yang sesuai. Proses ini ditangani oleh fungsi hapus yang akan mengubah nilai keanggotaan dari posting pada kategori terpilih menjadi 0. Fungsi ini akan dipanggil apabila admin menekan tombol 'hapus' pada posting tertentu di halaman dashboard. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.13.

Selain itu admin dapat menentukan nilai threshold Fuzzy yang digunakan dalam aplikasi. Nilai threshold Fuzzy ini digunakan untuk menentukan derajat nilai keanggotaan dari data yang masih diterima oleh sistem. Terdapat empat pilihan yang dapat dipilih oleh admin yaitu mengambil nilai keanggotaan tertinggi pada setiap data, mengambil dua nilai keanggotaan tertinggi, mengambil nilai threshold berdasarkan rata-rata nilai keanggotaan data pada setiap kategori, dan mengatur secara manual nilai threshold Fuzzy yang diinginkan. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 7.14.

#### **4.2.12. Kelas Atur\_posting**

Kelas ini menangani proses pengaturan konten hasil query model yang akan dikirimkan ke view dari aplikasi. Hasil query dari model berupa sejumlah objek dari kelas data. Objek ini tidak dapat ditampilkan langsung oleh view. Kelas ini membentuk string berupa tag HTML yang berisi data dari objek hasil query pada model sehingga ketika string tersebut dikirim ke view akan membentuk tampilan web dari data.

### **4.3. Implementasi Lapisan Data**

Lapisan ini hanya berhubungan dengan kelas-kelas pada lapisan kontrol. Lapisan data menyimpan atribut yang diperlukan

oleh kelas-kelas pada lapisan kontrol dan berisi atribut yang bersifat private.

#### **4.3.1. Kelas Data**

Kelas ini berisi atribut-atribut yang diperlukan untuk merepresentasikan sebuah posting pada back-end aplikasi. Atribut tersebut antara lain merepresentasikan nama pengguna yang menulis posting, ID posting, konten dari posting, tanggal posting dibuat, tanggal posting di-update, alamat link yang disertakan, dan alamat gambar yang disertakan. Kelas ini juga dilengkapi dengan fungsi-fungsi untuk mengakses atribut-atribut tersebut.

#### **4.3.2. Kelas GroupPost**

Kelas ini merepresentasikan kumpulan posting yang diatur berdasarkan penomoran halaman (paging). Atribut dari kelas ini adalah sebuah list yang berisi objek dari kelas Data dan juga paging dari list tersebut. Kelas ini juga dilengkapi dengan fungsi-fungsi untuk mengakses atribut-atribut tersebut.

#### **4.3.3. Kelas Post\_model**

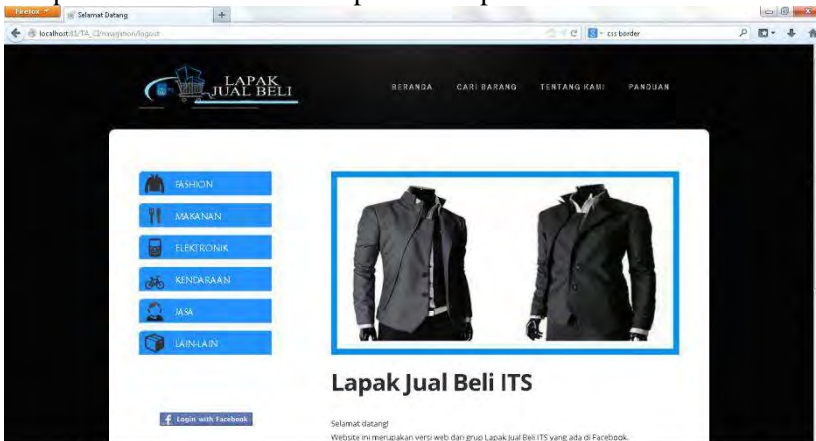
Kelas ini berisi atribut-atribut yang diperlukan untuk merepresentasikan sebuah posting pada front-end aplikasi. Atribut tersebut antara lain ID posting di basis data, ID posting dari Facebook, ID pengguna yang menulis posting, nama pengguna yang menulis posting, konten posting, dan konten yang telah melalui tahap pra-proses. Kelas ini juga dilengkapi dengan fungsi-fungsi untuk mengakses atribut-atribut tersebut.

### **4.4. Implementasi Antarmuka Pengguna**

Implementasi tampilan antarmuka pengguna dibuat dengan HyperText Markup Language (HTML) yang disertai dengan Cascading Style Sheet (CSS) dan JavaScript untuk menambah kesan interaktif pada website. Berikut ini akan dijelaskan mengenai implementasi tampilan antarmuka pengguna aplikasi.

#### 4.4.1. Tampilan Halaman Beranda

Halaman Beranda merupakan halaman pertama yang dilihat oleh pengguna ketika membuka aplikasi. Pada halaman ini terdapat logo dari aplikasi, menu utama aplikasi yang terletak pada bagian atas, menu samping yang berisi daftar kategori yang tersedia, photobucket yang menampilkan sejumlah gambar untuk mempercantik tampilan, serta deskripsi singkat mengenai aplikasi. Tampilan halaman beranda dapat dilihat pada Gambar 4.1.



**Gambar 4.1 Tampilan Halaman Beranda**

#### 4.4.2. Tampilan Halaman Pencarian Barang

Halaman Cari Barang digunakan untuk melakukan pencarian terhadap posting tertentu berdasarkan kata kunci dari pengguna. Pada halaman ini terdapat form pencarian yang terdiri atas sebuah textbox untuk memasukkan kata kunci dan sebuah tombol untuk mengirim perintah pencarian ke controller. Hasil pencarian akan ditampilkan pada panel di bawah form pencarian. Tampilan halaman pencarian barang dapat dilihat pada Gambar 4.2.

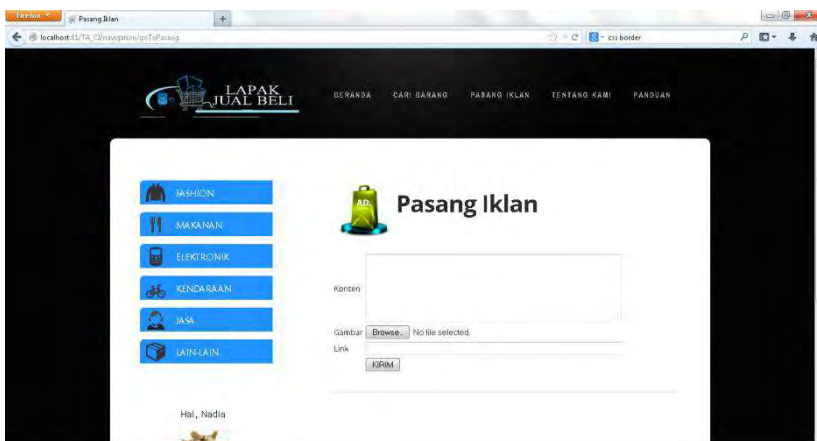
#### 4.4.3. Tampilan Halaman Pemasangan Iklan

Halaman Pasang Iklan digunakan sebagai media untuk menulis ke wall dari grup melalui aplikasi. Halaman ini hanya dapat diakses oleh pengguna yang telah login dengan akun Facebook. Pada

halaman ini terdapat form pencarian yang terdiri atas sebuah textarea sebagai tempat menulis konten iklan, sebuah file uploader untuk meng-upload gambar yang ingin disertakan pada iklan, sebuah textbox sebagai tempat menyertakan link pada iklan, dan sebuah tombol untuk mengirim perintah pengiriman konten iklan ke server Facebook. Tampilan halaman pemasangan iklan ditunjukkan pada Gambar 4.3.



**Gambar 4.2 Tampilan Halaman Pencarian Barang**



**Gambar 4.3 Tampilan Halaman Pemasangan Iklan**

#### 4.4.4. Tampilan Halaman Data Posting

Halaman data posting akan menampilkan data posting dari kategori terpilih. Daftar kategori yang dapat dipilih tersedia pada menu samping dari aplikasi. Posting akan ditampilkan berdasarkan tingkat relevansinya pada kategori terpilih yang merupakan nilai keanggotaan dari data posting tersebut terhadap kategori terkait. Tampilan halaman data posting ditunjukkan pada Gambar 4.4. Apabila pengguna ingin melihat detail dari posting tertentu, pengguna dapat memilih posting tersebut dan sistem akan mengarahkan pengguna ke halaman detail dari posting tersebut. Tampilan halaman detail data posting ditunjukkan pada Gambar 4.5.



Gambar 4.4 Tampilan Halaman Data Posting

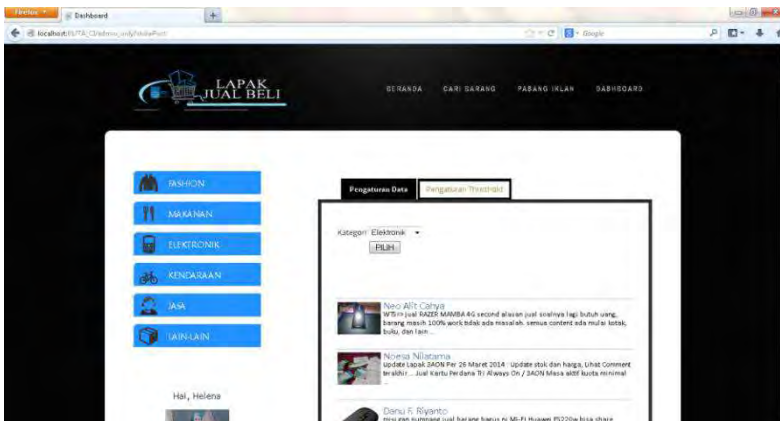
#### 4.4.5. Tampilan Halaman Dashboard Admin

Halaman Dashboard Admin menyediakan menu-menu yang hanya bisa diakses oleh admin aplikasi. Tampilan halaman Dashboard Admin ditunjukkan pada Gambar 4.6 dan Gambar 4.7. Pada dashboard admin terdapat dua menu utama yaitu menu Pengaturan Data yang digunakan untuk menghapus data posting dari kategori dimana posting tersebut tidak terkelompokkan pada kategori yang sesuai. Admin dapat memilih dari setiap kategori posting mana yang ingin dihapus dari kategori tersebut. Tampilan menu

Pengaturan Data ditunjukkan pada Gambar 4.6. Selain itu juga terdapat menu Pengaturan Threshold Fuzzy dimana admin dapat menentukan derajat nilai keanggotaan data yang masih diterima oleh sistem. Tampilan menu Pengaturan Threshold Fuzzy ditunjukkan pada Gambar 4.7.

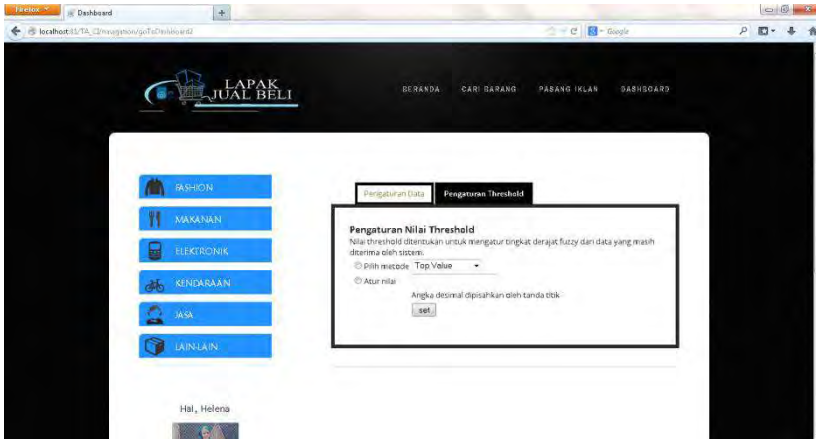


**Gambar 4.5 Tampilan Halaman Detail Data Posting**



**Gambar 4.6 Tampilan Halaman Pengaturan Data Posting pada Dashboard Admin**





**Gambar 4.7 Tampilan Halaman Pengaturan Threshold Fuzzy pada Dashboard Admin**

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan pengujian kegunaan sistem atau pengujian usability. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Pengujian kegunaan sistem dilakukan dengan mengetahui tanggapan dari pengguna terhadap kegunaan sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

#### **5.1. Lingkungan Pengujian**

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan alat kaku sebagai berikut:

Processor	: Intel® Core™ i3-380MProcessor (2.53 GHz, Cache 3 MB)
Memory	: 4.00 GB
Jenis Device	: Laptop
Sistem Operasi	: Microsoft Windows 7 Ultimate 32 bit
Browser	: Mozilla Firefox 26.0

#### **5.2. Skenario Pengujian**

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas, pengujian hasil pengelompokan, dan pengujian kegunaan sistem. Pengujian fungsionalitas menggunakan metode kotak hitam (black box). Metode ini menekankan pada kesesuaian hasil keluaran sistem. Pengujian terhadap hasil pengelompokan dilakukan dengan membandingkan hasil pelabelan kelas oleh sistem dengan hasil analisis manual yang dilakukan oleh pengembang aplikasi. Hasil perbandingan tersebut digunakan untuk mendapatkan nilai presisi dan recall. Pengujian kegunaan sistem dilakukan dengan melakukan survei pada pengguna aplikasi.

### 5.2.1. Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.4. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

#### 5.2.1.1. Pengujian Fitur Melihat Data Posting

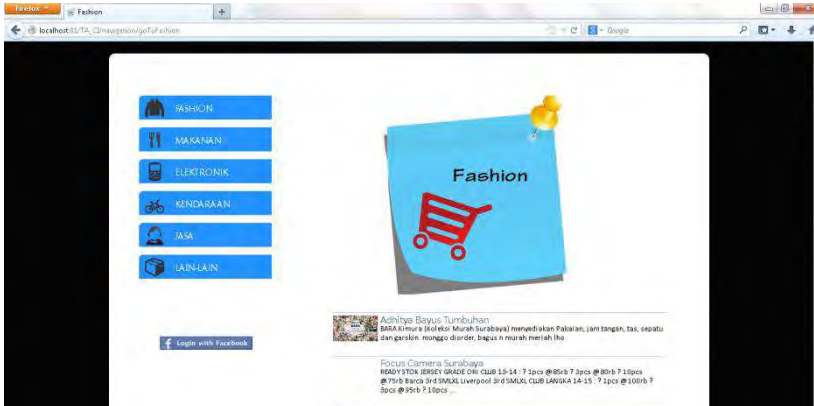
Pengujian fitur melihat data posting merupakan pengujian terhadap kemampuan aplikasi dalam menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna. Dalam pengujian fitur ini digunakan 100 buah data posting yang akan dikelompokkan ke dalam 6 kategori yaitu fashion, makanan, elektronik, kendaraan, jasa, dan lain-lain. Deskripsi mengenai data yang digunakan untuk pengujian dijelaskan pada Tabel 5.1. Rincian skenario kasus penggunaan melihat data posting untuk kategori fashion dapat dilihat pada Tabel 5.2.

**Tabel 5.1 Tabel Data Posting untuk Pengujian Fitur**

Kategori	Jumlah Data
Fashion	23
Makanan	3
Elektronik	50
Kendaraan	2
Jasa	12
Lain-lain	10
<b>Total</b>	<b>100</b>

**Tabel 5.2. Pengujian Fitur Melihat Data Posting Skenario 1**

<b>ID</b>	UJ.UC-0001
<b>Referensi Kasus Penggunaan</b>	UC-0001
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kategori fashion.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori fashion ditunjukkan pada Gambar 5.1. Contoh data posting kategori fashion dapat dilihat pada Gambar 5.2.



**Gambar 5.1 Hasil Uji Fitur Melihat Data Posting Kategori Fashion**

SALAM,

butuh dana nih jadi mau jual sepatu bola PUMA POWERCAT C-3  
ukuran 44 (EUR)

lengkap dengan box

kondisi masih mulus baru sekali pake walaupun udah lama belinya

harga bisa dinego

dengan bonus kaos kaki bola nike dri-fit (sampe lutut)

SMS 087851803379

siapa cepat dia duluan!

COD sekitaran ITS

**Gambar 5.2 Contoh Posting Kategori Fashion**

### 5.2.1.2. Pengujian Fitur Mencari Posting Tertentu

Pengujian fitur mencari posting tertentu merupakan pengujian terhadap kemampuan aplikasi dalam mencari posting yang bersesuaian dengan kata kunci yang diberikan oleh pengguna. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.3.

**Tabel 5.3 Pengujian Fitur Mencari Posting Tertentu Skenario 1**

<b>ID</b>	UJ.UC-0002
<b>Referensi Kasus Penggunaan</b>	UC-0002
<b>Nama</b>	Pengujian fitur mencari posting tertentu pada aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur mencari posting berdasarkan kata kunci dari pengguna.
<b>Skenario</b>	Posting yang bersesuaian dengan kata kunci dari pengguna akan ditampilkan pada panel hasil pencarian.
<b>Kondisi Awal</b>	Halaman Cari Barang dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kata kunci pencarian yaitu 'baju jersey'.
<b>Langkah Pengujian</b>	Pengguna memasukkan kata kunci pencarian dan memilih tombol 'cari'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan data posting yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan data posting yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan aplikasi menampilkan hasil pencarian ditunjukkan pada Gambar 5.3.



**Gambar 5.3 Hasil Uji Fitur Mencari Posting Tertentu Skenario 1**

### 5.2.1.3. Pengujian Fitur Mem-posting Iklan

Pengujian fitur mem-posting iklan merupakan pengujian terhadap kemampuan aplikasi dalam menulis pada wall grup melalui aplikasi. Dalam pengujian ini tidak digunakan grup Lapak Jual Beli ITS yang terdapat di Facebook karena dalam penulisan iklan di grup tersebut memerlukan proses persetujuan dari admin grup. Oleh karena itu digunakan grup uji coba bernama Coba Post. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.4.

**Tabel 5.4 Pengujian Fitur Mem-posting Iklan Skenario 1**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur mem-posting iklan melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur menulis pada wall grup melalui aplikasi.
<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi dan posting akan ditampilkan pada wall grup.

<b>Kondisi Awal</b>	Pengguna merupakan anggota grup dan telah login dengan akun Facebook. Halaman Pasang Iklan telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa konten teks ‘Tes aplikasi’ dan sebuah gambar sebagai lampiran.
<b>Langkah Pengujian</b>	Pengguna memasukkan konten iklan dan meng-upload gambar sebagai lampiran lalu memilih tombol ‘kirim’.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting baru ditampilkan pada wall grup uji coba.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting baru ditampilkan pada wall grup uji coba.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan respon berhasil dari sistem ditunjukkan pada Gambar 5.4. Tampilan data posting baru pada wall grup Facebook ditunjukkan pada Gambar 5.5.



**Gambar 5.4 Hasil Uji Fitur Mem-posting Iklan Skenario 1**





**Gambar 5.5 Tampilan Data Posting Baru pada Wall Grup**

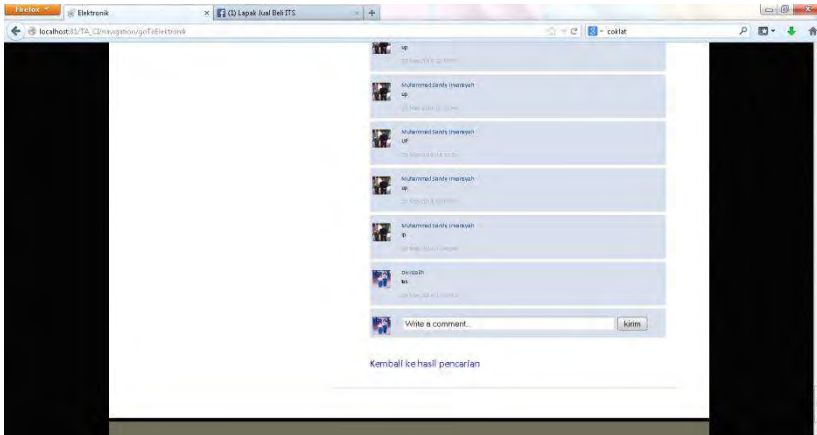
#### 5.2.1.4. Pengujian Fitur Menulis Komentar pada Posting

Pengujian fitur menulis komentar pada posting merupakan pengujian terhadap kemampuan aplikasi dalam menulis komentar pada posting melalui aplikasi. Fitur ini hanya tersedia untuk pengguna yang merupakan anggota dari grup Lapak Jual Beli ITS di Facebook yang melakukan login di aplikasi. Rincian skenario kasus penggunaan ini dapat dilihat pada Tabel 5.5.

**Tabel 5.5 Pengujian Fitur Menulis Komentar pada Posting Skenario 1**

<b>ID</b>	UJ.UC-0004
<b>Referensi Kasus Penggunaan</b>	UC-0004
<b>Nama</b>	Pengujian fitur menulis komentar pada posting melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur menulis komentar pada posting melalui aplikasi.
<b>Skenario</b>	Komentar yang ditulis oleh pengguna akan ditambahkan pada tab komentar dari posting.
<b>Kondisi Awal</b>	Pengguna merupakan anggota dari grup dan telah login dengan akun Facebook. Pengguna telah memilih posting yang ingin diberi komentar.

<b>Data Uji</b>	Data uji berupa konten komentar yang ingin diberikan dan posting yang ingin diberi komentar.
<b>Langkah Pengujian</b>	Pengguna memasukkan konten komentar pada textbox yang tersedia dan memilih tombol 'kirim'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem dan menambahkan komentar di tab komentar dari posting.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem dan menambahkan komentar di tab komentar dari posting.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan komentar pengguna di tab komentar dari posting ditunjukkan pada Gambar 5.6.



**Gambar 5.6 Hasil Uji Fitur Menulis Komentar pada Posting Skenario 1**

### 5.2.1.5. Pengujian Fitur Memberikan Like pada Posting

Pengujian fitur memberikan like pada posting merupakan pengujian terhadap kemampuan aplikasi dalam memberikan like pada posting melalui aplikasi. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.6.

**Tabel 5.6 Pengujian Fitur Memberikan Like pada Posting Skenario 1**

<b>ID</b>	UJ.UC-0005
<b>Referensi Kasus Penggunaan</b>	UC-0005
<b>Nama</b>	Pengujian fitur memberikan like pada posting melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur memberikan like pada posting melalui aplikasi.
<b>Skenario</b>	Penanda bahwa posting telah diberikan like ditampilkan pada tab like dari posting.
<b>Kondisi Awal</b>	Pengguna telah login dengan akun Facebook dan telah memilih posting yang ingin diberi like.
<b>Data Uji</b>	Data uji berupa posting yang ingin diberi like.
<b>Langkah Pengujian</b>	Pengguna memilih pilihan 'Suka' pada posting.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem dan menampilkan tulisan 'Disukai' pada tab like dari posting.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem dan menampilkan tulisan 'Disukai' pada tab like dari posting.
<b>Hasil Pengujian</b>	Berhasil

<b>Kondisi Akhir</b>	Tampilan tab like dari posting terpilih ditunjukkan pada Gambar 5.7.
----------------------	--



Gambar 5.7 Hasil Uji Fitur Memberikan Like pada Posting Skenario 1

### 5.2.1.6. Pengujian Fitur Menghapus Posting dari Kategori

Pengujian fitur menghapus posting dari kategori merupakan pengujian terhadap kemampuan aplikasi dalam menghapus posting yang tidak dikelompokkan ke kategori yang sesuai dari posting tersebut. Fitur ini hanya bisa diakses oleh admin aplikasi. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Pengujian Fitur Menghapus Posting dari Kategori

<b>ID</b>	UJ.UC-0006
<b>Referensi Kasus Penggunaan</b>	UC-0006
<b>Nama</b>	Pengujian fitur menghapus posting dari kategori pada aplikasi.

<b>Tujuan Pengujian</b>	Menguji fitur menghapus posting yang tidak dikelompokkan pada kategori yang sesuai dari posting tersebut.
<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi dan posting akan dihapus dari kategori.
<b>Kondisi Awal</b>	Admin telah login dengan akun Facebook dan halaman Dashboard menu Pengaturan Data telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa posting dengan ID Data 58 pada kategori Jasa.
<b>Langkah Pengujian</b>	Admin memilih kategori Jasa dan memilih posting dengan ID Data 58 kemudian memilih tombol 'hapus'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting dihapus dari daftar posting kategori.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting dihapus dari daftar posting kategori.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Perbandingan tampilan halaman data posting sebelum dan sesudah posting dihapus ditunjukkan pada Gambar 5.8 dan Gambar 5.9

### 5.2.1.7. Pengujian Fitur Mengatur Nilai Threshold Fuzzy

Pengujian fitur mengatur nilai threshold Fuzzy merupakan pengujian terhadap kemampuan aplikasi dalam mengatur nilai threshold Fuzzy berdasarkan metode yang dipilih oleh pengguna. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.8.



**Gambar 5.8 Halaman Data Posting Sebelum Posting Dihapus**



**Gambar 5.9 Halaman Data Posting Setelah Posting Dihapus**

**Tabel 5.8 Pengujian Fitur Mengatur Nilai Threshold Fuzzy**

ID	UJ.UC-0007
Referensi Kasus Penggunaan	UC-0007
Nama	Pengujian fitur mengatur nilai threshold Fuzzy.

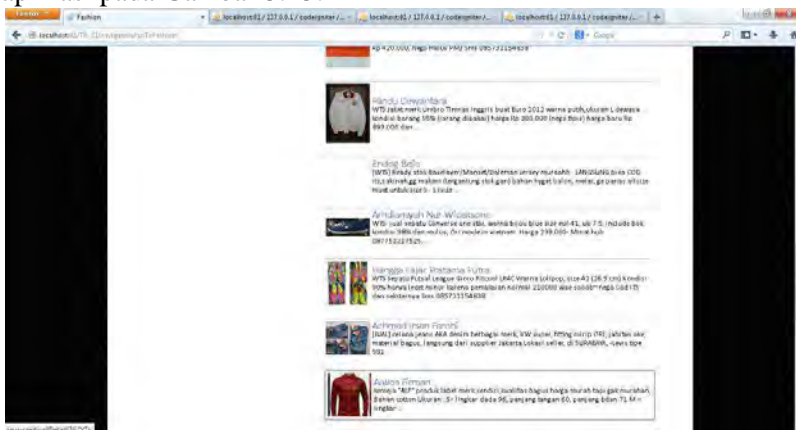
<b>Tujuan Pengujian</b>	Menguji fitur mengatur nilai threshold Fuzzy berdasarkan metode yang dipilih oleh pengguna.
<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi dan hanya posting dengan nilai keanggotaan pada kategori di atas threshold yang akan ditampilkan.
<b>Kondisi Awal</b>	Admin telah login dengan akun Facebook dan halaman Dashboard menu Pengaturan Threshold telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa metode pengambilan nilai keanggotaan tertinggi dan posting dengan ID Data 76 dengan nilai keanggotaan pada kategori fashion sebesar 0.180237 dan pada kategori makanan sebesar 0.046398
<b>Langkah Pengujian</b>	Admin memilih metode pengambilan nilai keanggotaan tertinggi sebagai metode pengaturan nilai threshold Fuzzy dan memilih tombol 'set'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting hanya ditampilkan pada kategori fashion.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting hanya ditampilkan pada kategori fashion.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Perbandingan tampilan halaman data posting kategori fashion dan makanan ditunjukkan pada Gambar 5.10 dan Gambar 5.11.

Rincian skenario pengujian fitur dijelaskan pada Lampiran B. Hasil Pengujian

### 5.2.2. Pengujian Kegunaan Sistem

Pengujian kegunaan sistem dilakukan dengan cara melakukan survei pada pengguna dari aplikasi. Pengguna diminta untuk menguji fitur-fitur yang terdapat pada aplikasi kemudian diminta untuk mengisi kuisioner pendapat pengguna tentang aplikasi. Dalam pengujian kegunaan ini terdapat 20 orang responden. Pengujian ditekankan pada tingkat kegunaan aplikasi dalam

membantu pencarian informasi, tingkat akurasi pengelompokan, tingkat responsivitas aplikasi, dan penilaian fitur aplikasi. Detail mengenai kuisisioner pengguna dan profil responden dapat dilihat pada Lampiran B. Hasil Pengujian. Hasil survei tingkat kegunaan aplikasi dalam pencarian informasi ditunjukkan grafik pada Gambar 5.12, tingkat akurasi pengelompokan pada Gambar 5.13, tingkat responsivitas aplikasi pada Gambar 5.14, dan tentang penilaian fitur aplikasi pada Gambar 5.15.

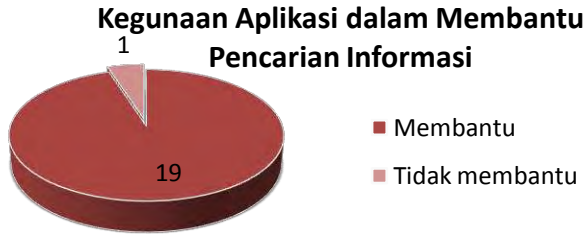


**Gambar 5.10 Tampilan Halaman Data Posting Kategori Fashion**

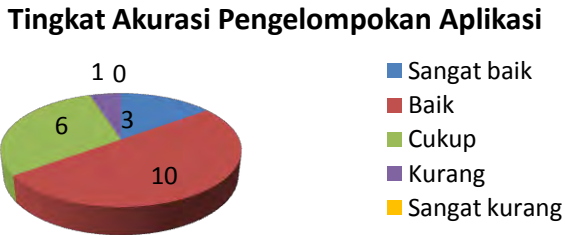


**Gambar 5.11 Tampilan Halaman Data Posting Kategori Makanan**

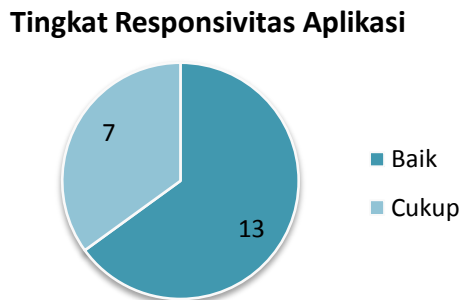




**Gambar 5.12 Grafik Hasil Survei Tingkat Kegunaan Aplikasi dalam Membantu Pencarian Informasi**



**Gambar 5.13 Grafik Hasil Survei Tingkat Akurasi Pengelompokan Aplikasi**



**Gambar 5.14 Grafik Hasil Survei Tingkat Responsivitas Aplikasi**



**Gambar 5.15 Grafik Hasil Survei Penilaian Fitur Aplikasi**

### 5.3. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional, evaluasi hasil pengelompokan, dan evaluasi kegunaan sistem.

#### 5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.9. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

**Tabel 5.9 Rangkuman Hasil Pengujian**

ID	Nama	Skenario	Hasil
UJ.UC-0001	Melihat data posting.	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
		Skenario 6	Berhasil

ID	Nama	Skenario	Hasil
UJ.UC-0002	Mencari posting tertentu	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0003	Mem-posting iklan	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ.UC-0004	Menulis komentar pada posting	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0005	Memberikan like pada posting	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0006	Menghapus posting dari kategori		Berhasil
UJ.UC-0007	Mengatur nilai threshold Fuzzy		Berhasil

### 5.3.2. Evaluasi Hasil Pengelompokan

Perhitungan presisi dan recall dilakukan untuk mengukur performa dari sistem. Presisi dihitung dari jumlah informasi relevan yang berhasil dideteksi dibandingkan dengan seluruh hasil deteksi. Sedangkan recall didapatkan dari perbandingan jumlah informasi relevan yang berhasil dideteksi dengan jumlah data relevan yang sebenarnya yang didapat dari analisis pengembang aplikasi. Dalam aplikasi ini, terdapat tujuh mode pengaturan threshold Fuzzy yang digunakan. Masing-masing nilai threshold memberikan hasil perhitungan yang berbeda. Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold pengambilan nilai tertinggi dapat dilihat pada Tabel 5.10.

**Tabel 5.10 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Top Value**

Kategori	Retrieve	Relevant	Relevant Retrieve	Presisi	Recall
Fashion	16	26	15	0,94	0,58
Makanan	3	3	3	1	1

<b>Elektronik</b>	50	58	50	1	0,86
<b>Kendaraan</b>	11	3	3	0,27	1
<b>Jasa</b>	17	13	10	0,59	0,77
<b>Lain-lain</b>	3	7	3	1	0,43
<b>Rata-rata</b>				0,80	0,77

Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold pengambilan dua nilai tertinggi dapat dilihat pada Tabel 5.11.

**Tabel 5.11 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Top 2 Value**

<b>Kategori</b>	<b>Retrieve</b>	<b>Relevant</b>	<b>Relevant Retrieve</b>	<b>Presisi</b>	<b>Recall</b>
<b>Fashion</b>	22	26	18	0,82	0,69
<b>Makanan</b>	6	3	3	0,5	1
<b>Elektronik</b>	63	58	55	0,87	0,95
<b>Kendaraan</b>	22	3	3	0,14	1
<b>Jasa</b>	31	13	12	0,39	0,92
<b>Lain-lain</b>	3	7	3	1	0,43
<b>Rata-rata</b>				0,62	0,83

Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold pengambilan nilai di atas rata-rata nilai keanggotaan data dapat dilihat pada Tabel 5.12.

**Tabel 5.12 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Average Value**

<b>Kategori</b>	<b>Retrieve</b>	<b>Relevant</b>	<b>Relevant Retrieve</b>	<b>Presisi</b>	<b>Recall</b>
<b>Fashion</b>	28	26	23	0,82	0,88
<b>Makanan</b>	6	3	3	0,5	1
<b>Elektronik</b>	62	58	55	0,89	0,95
<b>Kendaraan</b>	20	3	3	0,15	1
<b>Jasa</b>	28	13	12	0,43	0,92
<b>Lain-lain</b>	3	7	3	1	0,43
<b>Rata-rata</b>				0,63	0,86

Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold berdasarkan input admin dengan nilai threshold sebesar 0,075 dapat dilihat pada Tabel 5.13.

**Tabel 5.13 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Input Admin 1**

Kategori	Retrieve	Relevant	Relevant Retrieve	Presisi	Recall
Fashion	24	26	21	0,88	0,81
Makanan	6	3	3	0,5	1
Elektronik	59	58	54	0,92	0,93
Kendaraan	17	3	3	0,18	1
Jasa	20	13	10	0,5	0,77
Lain-lain	3	7	3	1	0,43
<b>Rata-rata</b>				0,66	0,82

Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold berdasarkan input admin dengan nilai threshold sebesar 0,1 dapat dilihat pada Tabel 5.14.

**Tabel 5.14 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Input Admin 2**

Kategori	Retrieve	Relevant	Relevant Retrieve	Presisi	Recall
Fashion	20	26	19	0,95	0,73
Makanan	3	3	3	1	1
Elektronik	56	58	53	0,95	0,91
Kendaraan	13	3	3	0,23	1
Jasa	19	13	10	0,53	0,77
Lain-lain	3	7	3	1	0,43
<b>Rata-rata</b>				0,78	0,81

Hasil perhitungan tingkat presisi dan recall dari sistem dengan menggunakan mode pengaturan threshold berdasarkan input admin dengan nilai threshold sebesar 0,125 dapat dilihat pada Tabel 5.15. Hasil perhitungan tingkat presisi dan recall dari sistem dengan

menggunakan mode pengaturan threshold berdasarkan input admin dengan nilai threshold sebesar 0,2 dapat dilihat pada Tabel 5.16.

**Tabel 5.15 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Input Admin 3**

Kategori	Retrieve	Relevant	Relevant Retrieve	Presisi	Recall
Fashion	16	26	15	0,94	0,58
Makanan	3	3	3	1	1
Elektronik	52	58	50	0,96	0,86
Kendaraan	12	3	3	0,25	1
Jasa	14	13	9	0,64	0,69
Lain-lain	3	7	3	1	0,43
<b>Rata-rata</b>				0,80	0,76

**Tabel 5.16 Hasil Perhitungan Presisi dan Recall dengan Threshold Mode Input Admin 4**

Kategori	Retrieve	Relevant	Relevant Retrieve	Presisi	Recall
Fashion	7	26	7	1	0,27
Makanan	3	3	3	1	1
Elektronik	35	58	35	1	0,60
Kendaraan	5	3	3	0,6	1
Jasa	7	13	6	0,86	0,46
Lain-lain	3	7	3	1	0,43
<b>Rata-rata</b>				0,91	0,63

Kesalahan pemberian label kelas kategori pada data sering kali disebabkan konten dari data uji mengandung kata yang merupakan ciri spesifik dari kategori lain. Contoh data uji yang memiliki kecenderungan salah dalam pelabelan data ditunjukkan pada Gambar 5.16. Pada posting tersebut barang yang ditawarkan adalah tas dengan model sayap vespa. Seharusnya posting tersebut dikelompokkan ke kategori fashion. Akan tetapi karena dalam konten tersebut terdapat kata yang merupakan salah satu ciri spesifik dari kategori kendaraan yaitu vespa dan kata tersebut memiliki frekuensi kemunculan yang lebih tinggi jika dibandingkan dengan

kata tas, maka data posting tersebut memiliki nilai keanggotaan pada kategori kendaraan lebih tinggi dari pada nilai keanggotaan pada kategori fashion. Di sini peran dari nilai threshold Fuzzy berperan untuk menentukan pelabelan kelas pada data. Apabila mode threshold Fuzzy yang digunakan adalah pengambilan nilai keanggotaan tertinggi, maka tentunya posting ini tidak akan muncul di kategori fashion. Namun jika menggunakan mode yang lain, misal mode pengambilan nilai di atas rata-rata maka posting ini dapat masuk ke dalam kategori fashion.

Khusus untuk rekan2 sesama vespa mania, atau scooter mania. Belum lengkap rasanya kalo riding/nongkrong bersama motor kesayangan tapi belum pake Tas slempang model sayap depan vespa & tempong tutup mesin vespa classic. Cocok juga untuk yang hobby koleksi tas unik..

Dijamin makin keren saat riding, nongkrong, dan jalan2 (pengalaman saya sendiri kalo pas make kemana2 hehe.. )

100% hand made, made in Indonesia.

Available berbagai warna so grab fast bro..

price: 240rb aja

Untuk info2 lebih lanjut silahkan dengan senang hati sms ke nomer saya di 08977257552

Melayani COD area mulyosari..

Terima kasih.. Salam vespa mania..

**Gambar 5.16 Contoh Posting yang Memiliki Kecenderungan Salah Pelabelan Data**

Namun di sisi lain penggunaan metode Fuzzy ini tidak selalu memberikan dampak negatif. Karena konsep dari klasifikasi Fuzzy dimana data dapat memiliki lebih dari satu label kelas, maka bila diimplementasikan pada data yang sesuai maka metode ini dapat mendeteksi informasi yang kurang dominan pada suatu data. Seperti contoh posting pada Gambar 5.17 dimana inti utama dari barang yang ditawarkan adalah jasa berupa penyewaan barang. Namun dengan menggunakan metode klasifikasi Fuzzy, maka informasi elektronik juga dapat ditangkap dan kemudian direpresentasikan dalam bentuk nilai keanggotaan data tidak hanya terhadap kategori

jasa namun juga nilai keanggotaan data terhadap kategori elektronik, tentu saja dengan nilai keanggotaan yang berbeda tergantung dari informasi yang lebih dominan yang terdapat pada data.

disewakan (sewa)  
 kamera digital pocket  
 canon powershoot a550  
 blitz  
 optical zoom  
 digital zoom  
 7.1 mega pixel  
 micro suport  
 video suport  
 mmc 2gb include  
 catu daya:  
 2x batrai AA exlude (tidak termasuk)  
 syarat:  
 status pemohon mahasiswa aktif,  
 fotokopi ktp atau ktm.  
 harga sewa 1x24 jam (15rb)  
 085730405021 (iman)  
 keputih gang.3 no.30  
 cocok untuk dokumentasi acara jurusan, wisuda, seminar, himpunan, traveling,  
 dll.

**Gambar 5.17 Contoh Posting 2**

Dari hasil perhitungan presisi dan recall dapat dihitung nilai F Measure. Nilai F Measure dapat digunakan untuk menggambarkan kualitas performa dari sistem. Tabel 5.17 menunjukkan nilai F Measure untuk pengelompokan posting dengan tujuh mode pengaturan threshold Fuzzy.

Berdasarkan hasil perhitungan nilai F Measure yang ditunjukkan pada Tabel 5.17, diketahui bahwa (dengan melakukan pengecualian pada mode input admin) mode pengambilan nilai keanggotaan tertinggi memberikan performa dengan nilai yang



paling baik. Hal ini berkaitan dengan salah satu kelemahan metode klasifikasi Fuzzy yaitu tingkat sensitivitasnya terhadap informasi yang kurang dominan pada data sehingga seringkali berakibat pada kesalahan dalam pemberian label kelas pada data. Mode ini hampir mirip dengan pemberian nilai threshold yang cukup tinggi pada pengujian. Sehingga mode ini akan mengeliminasi detail-detail kecil yang ada pada informasi dan hanya mengambil informasi yang dominan saja. Oleh karena itu performa dengan menggunakan mode ini memberikan hasil yang lebih baik.

**Tabel 5.17 Daftar Hasil Perhitungan Nilai F Measure pada Pengujian Pengelompokan Posting**

<b>Mode Threshold</b>	<b>F Measure</b>
<b>Top Value</b>	0,7860
<b>Top 2 Value</b>	0,71
<b>Average Value</b>	0,7295
<b>Input Admin, t = 0,075</b>	0,7331
<b>Input Admin, t = 0,1</b>	0,7910
<b>Input Admin, t = 0,125</b>	0,7788
<b>Input Admin, t = 0,2</b>	0,7424

Sedangkan untuk pengaturan nilai threshold berdasarkan input admin, dapat dilihat pola bahwa dengan penambahan nilai threshold Fuzzy, maka berbanding lurus dengan kenaikan nilai F Measure. Akan tetapi pada range nilai threshold antara 0,1 hingga 0,125 terdapat titik balik dimana jika nilai threshold terus ditambahkan maka akan berdampak pada nilai F Measure yang semakin menurun. Pemberian nilai threshold yang tinggi memang dapat mengeliminasi detail-detail informasi yang kurang dominan, akan tetapi jika nilai threshold terlalu tinggi, maka akan semakin banyak informasi yang tidak tertangkap dan tentunya berpengaruh pada kualitas performa dari sistem. Berdasarkan observasi ini, maka untuk kasus pengujian ini, nilai optimal dari threshold Fuzzy terdapat pada range 0,1 hingga 0,125.

### **5.3.3. Evaluasi Pengujian Kegunaan Sistem**

Berdasarkan survei yang dilakukan terhadap 20 responden, didapatkan bahwa 19 responden atau sekitar 95% responden berpendapat bahwa aplikasi ini dapat membantu mempermudah pengguna dalam mencari informasi tertentu dengan sekitar 65% responden berpendapat bahwa tingkat responsivitas aplikasi baik dan 60% responden berpendapat bahwa fitur yang ada pada aplikasi sudah baik. Akan tetapi pada penilaian tingkat akurasi pengelompokan hanya 50% responden yang berpendapat bahwa tingkat akurasi sistem baik, sedangkan 30% responden berpendapat bahwa tingkat akurasi sistem cukup. Hal ini dikarenakan terdapat sejumlah data yang tidak dikelompokkan pada kategori yang sesuai dengan penyebab yang telah dijelaskan pada subbab sebelumnya.

Oleh karena itu dapat disimpulkan bahwa tujuan dari aplikasi ini yaitu untuk membantu mempermudah pengguna dalam mencari informasi telah tercapai.

## LAMPIRAN A.KODE SUMBER

```
public class MySQLAccess {
    private Connection connect = null;
    private Statement statement = null;
    private PreparedStatement preparedStatement = null;
    private ResultSet resultSet = null;

    public MySQLAccess() {
    }

    public void init() throws SQLException
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connect =
DriverManager.getConnection("jdbc:mysql://localhost/codeigniter?"
                + "user=root&password=");
        } catch (ClassNotFoundException ex) {

Logger.getLogger(MySQLAccess.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public HashSet readPost(int id_kategori) throws
SQLException
    {
        statement = connect.createStatement();
        resultSet = statement.executeQuery("SELECT
POST.KONTEN_STEMMED FROM POST INNER JOIN KATEGORIDATA WHERE
POST.IS_VALIDATION IS NULL AND
KATEGORIDATA.ID_KATEGORI="+id_kategori+" AND
POST.ID_DATA=KATEGORIDATA.ID_DATA");
        HashSet words = new HashSet();
        while(resultSet.next())
        {
            String[] kata_stem = null;
            String kata =
resultSet.getString("konten_stemmed");
            kata_stem = kata.split(",");
            int ukuran = kata_stem.length;
            int j = 0;
            while(j<ukuran)
            {
                words.add(kata_stem[j]);
            }
        }
    }
}
```

```

        j++;
    }
}
return words;
}

public int readStemmed(int id_data, String param) throws
SQLException
{
    statement = connect.createStatement();
    resultSet = statement.executeQuery("SELECT * FROM
CODEIGNITER.POST WHERE ID_DATA="+id_data);
    int count = 0;
    while(resultSet.next())
    {
        String stem =
resultSet.getString("konten_stemmed");
        String[] pecah = stem.split(",");
        for(int i=0;i<pecah.length;i++)
        {
            if(pecah[i].equals(param))
                count++;
        }
    }
    return count;
}
. . .
}

```

**Kode Sumber 7.1 Kelas MySQLAccess**

```

public class PostGrabber {

    private String jeson;
    private String sejak;
    private String sampai;
    private GroupPost posting;

    public PostGrabber() {
        jeson = "";
        sejak = "";
        sampai = "";
        posting = new GroupPost();
    }

    public String getJeson() {
        return jeson;
    }
}

```

```

    }

    public GroupPost getPosting() {
        return posting;
    }

    public String getSampai() {
        return sampai;
    }

    public String getSejak() {
        return sejak;
    }

    public void getPost() throws MalformedURLException,
    IOException
    {
        URL grupfb = new
        URL("https://graph.facebook.com/343743608982496/feed?access_
        token=617854334919295|AT5j8d_o6E5PMf1CNuwFsMVQGKU");
        URLConnection con = grupfb.openConnection();
        BufferedReader in = new BufferedReader(new
        InputStreamReader(con.getInputStream()));
        String respon;
        while((respon = in.readLine()) != null)
            jeson+=respon;
        in.close();
    }

    public void decodeJson()
    {
        Gson gson = new Gson();
        posting = gson.fromJson(jeson, GroupPost.class);
    }

    public void setUnixSince()
    {
        String prev =
        posting.getPaging().getPrevious().toString();
        String[] pecahh = prev.split("since");
        int idx_dans_1 = pecahh[1].indexOf("&");
        sejak = pecahh[1].substring(1, idx_dans_1);
    }

    public void setUnixUntil()
    {
        String nex =
        posting.getPaging().getNext().toString();
        String[] pecah = nex.split("until");
    }

```

```

        int idx_dan_1 = pecah[1].indexOf("&");
        sampai = pecah[1].substring(1, idx_dan_1);
    }
}

```

### Kode Sumber 7.2 Kelas PostGrabber

```

public class Stemmer {

    private String dokumen;
    private String hasil_stem;

    public Stemmer() {
        dokumen = "";
        hasil_stem = "";
    }

    public String getDokumen() {
        return dokumen;
    }

    public void setDokumen(String dokumen) {
        this.dokumen = dokumen;
    }

    public String getHasil_stem() {
        return hasil_stem;
    }

    public void setHasil_stem(String hasil_stem) {
        this.hasil_stem= hasil_stem;
    }

    public void stemming() throws IOException, SQLException
    {
        if(dokumen!=null)
        {
            dokumen = dokumen.replaceAll("[^\\p{L}\\p{Nd}]",
" ");

            String kata[] = dokumen.split(" ");
            MySQLAccess dao = new MySQLAccess();
            dao.init();
            List<String> lis = new
ArrayList<String>(Arrays.asList(kata));
            lis.removeAll(Arrays.asList(" "));
            lis.removeAll(Arrays.asList(""));
            kata = lis.toArray(new String[lis.size()]);

            for(int i=kata.length-1;i>=0;i--)

```

```

        {
            if((kata[i].equalsIgnoreCase("bb")           ||
kata[i].equalsIgnoreCase("blackberry")              ||
kata[i].equalsIgnoreCase("bbm"))    && (i-1)>=0    && kata[i-1].equalsIgnoreCase("pin"))
                kata[i] = "";

            if((kata[i].equalsIgnoreCase("hp")         ||
kata[i].equalsIgnoreCase("handphone"))              &&
((i+1)<=(kata.length-1))    && (kata[i+1].matches(".*\\d+.*"))
|| kata[i+1].equalsIgnoreCase("wa")                 ||
kata[i+1].equalsIgnoreCase("whatsapp")              ||
kata[i+1].equalsIgnoreCase("line")))
                kata[i] = "";

        }

        for(int i=0;i<kata.length;i++)
        {
            //String temp =
kata[i].replaceAll("[^\\p{L}\\p{Nd}]", " ");
            String temp = kata[i].toLowerCase();

            Pattern p = Pattern.compile("-?\\d+");
            Matcher m = p.matcher(temp);
            temp = dao.normalisasi(temp);
            if(m.find())
                continue;
            System.out.println(temp);
            temp = NaziefAdriani(temp);
            System.out.println("hasil stem= "+temp);
            if(temp.length()<3)
                continue;

            if(dao.getTipeKataDasar(temp).equalsIgnoreCase("adjektiva")
|| dao.getTipeKataDasar(temp).equalsIgnoreCase("adverbia")
||

            dao.getTipeKataDasar(temp).equalsIgnoreCase("preposisi") ||
            dao.getTipeKataDasar(temp).equalsIgnoreCase("interjeksi") ||

            dao.getTipeKataDasar(temp).equalsIgnoreCase("konjungsi") ||
            dao.getTipeKataDasar(temp).equalsIgnoreCase("pronomina") ||

            dao.getTipeKataDasar(temp).equalsIgnoreCase("lain-lain") ||
            dao.getTipeKataDasar(temp).equalsIgnoreCase("numeralia"))
                continue;

            if(!dao.findWords(temp))
            {

```

```

        hasil_stem += temp;
        hasil_stem += ",";
    }
}
}
}

```

### Kode Sumber 7.3 Kelas Stemmer

```

public class DataPartition {
    private List<List<Integer>> partisi;

    public DataPartition() {
        partisi = new ArrayList<List<Integer>>();
    }

    public List<List<Integer>> getPartisi ()
    {
        return partisi;
    }

    public void setPartisi() throws SQLException
    {
        MySQLAccess dao = new MySQLAccess();
        dao.init();
        dao.hapusValidData();
        for(int i=0;i<5;i++)
        {
            int j = i+1;
            Object[] doc = new Object[50];
            doc = dao.readDokumen(j);
            int part;
            part = doc.length/10;
            if(part > 0)
            {
                int c = 0;
                for(int k=0;k<10;k++)
                {
                    List<Integer> temp = new
ArrayList<Integer>();
                    for(int m=0;m<part;m++)
                    {
                        temp.add((Integer)doc[c]);
                        c++;
                    }
                    if(k==0)
                    {

```



```

                for(int m=0;m<(doc.length%10);m++)
                {
                    temp.add((Integer)doc[c]);
                    c++;
                }
            }
            partisi.add(temp);
        }
    }
}

public void setValidationData(int round) throws
SQLException
{
    MySQLAccess dao = new MySQLAccess();
    dao.init();
    dao.hapusValidData();
    for(int i=0;i<50;i+=10)
    {
        int j = i+round;
        List<Integer> valData = new
ArrayList<Integer>();
        valData = partisi.get(j);
        for(int k=0;k<valData.size();k++)
        {
            int data = valData.get(k);
            dao.setValidData(data);
        }
    }
}
}

```

#### Kode Sumber 7.4 Kelas DataPartition

```

public class Training {

    private MySQLAccess dao;
    private int bagOfWords[][];
    private float bagOfWords2[][];
    private Object dokumen[];
    private Object kata[];
    private float tf[][];
    private float idf[];
    private int id_kt;
    private List<String> identitasKategori;
    private String tambahID[][] = {

```

```

        {"kaos", "jersey", "sepatu", "tas",
"jam", "sandal", "jaket", "polo", "jeans", "batik", "baju",
""},
        {"madu", "nasi", "menu", "makanan",
"susu", "minuman", "obat", "diet", "cokelat", "rasa", "",
""},
        {"ponsel", "laptop", "game", "perdana",
"harddisk", "baterai", "modem", "flashdisk", "internet",
"printer", "powerbank", "garansi"},
        {"motor", "mobil", "toyota", "suzuki",
"honda", "yamaha", "sepeda", "vespa", "", "", "", ""},
        {"sewa", "tiket", "servis", "recovery",
"jasa", "", "", "", "", "", "", ""}
    };

    public Training() {
        dao = new MySQLAccess();
        bagOfWords = new int[5000][5000];
        bagOfWords2 = new float[5000][5000];
        tf = new float[5000][5000];
        idf = new float[5000];
        identitasKategori = new ArrayList<String>();
        dokumen = new Object[100];
        kata = new Object[5000];
    }

    public int[][] getBagOfWords() {
        return bagOfWords;
    }

    public void setBagOfWords(int id_kategori) throws
SQLException, IOException {
        id_kt = id_kategori;
        dao.init();
        dokumen = dao.readDokumen(id_kategori);
        System.out.println("dokumen=" +dokumen.length);
        /*for(int i=0;i<dokumen.length;i++)
            System.out.println("doc id =
"+dokumen[i].toString());*/

        HashSet words = new HashSet();
        words = dao.readPost(id_kategori);
        /*Iterator<String> itr = words.iterator();
        while(itr.hasNext())
            System.out.println(itr.next());*/
        kata = words.toArray();
        System.out.println("kata= "+kata.length);
        /*for(int i=0;i<kata.length;i++)
            System.out.println(i+" "+kata[i].toString());*/
    }

```

```

        FileWriter write = new
FileWriter("D:/bagOfWords.txt");
        PrintWriter print_line = new PrintWriter(write);
        print_line.printf("TF%n");
        for (int i=0;i<dokumen.length;i++)
        {
            //int max = 0;
            int totalwords = 0;
            for (int j=0;j<kata.length;j++)
            {
                //perhitungan TF
                int a = dao.readStemmed((Integer)dokumen[i],
kata[j].toString());
                bagOfWords[i][j] = a;
                /*if (bagOfWords[i][j] > max)
                    max = a;*/
                totalwords += a;
                //print_line.printf("%s ",bagOfWords[i][j]);
            }

            for (int j=0;j<kata.length;j++)
            {
                tf[i][j] =
(float)bagOfWords[i][j]/(float)totalwords;
                print_line.printf("%s ",tf[i][j]);
            }
            print_line.printf("%n");
        }
        print_line.printf("%n");

        //perhitungan DF
        int totalPerTerm[] = new int[5000];
        print_line.printf("DF%n");
        for (int j=0;j<kata.length;j++)
        {
            int jumlah = 0;
            for (int i=0;i<dokumen.length;i++)
            {
                jumlah += bagOfWords[i][j];
            }
            totalPerTerm[j] = jumlah;
            print_line.printf("%s ", totalPerTerm[j]);
        }
        print_line.printf("%n");

        //perhitungan IDF
        print_line.printf("%nIDF%n");

```

```

        for(int j=0;j<kata.length;j++)
        {
            idf[j] = (float)
(Math.log((float)dokumen.length/(float)totalPerTerm[j])/Math
.log(2));
            print_line.printf("%s ", idf[j]);
        }
        print_line.close();
    }

    public float[][] getBagOfWords2() {
        return bagOfWords2;
    }

    public void setBagOfWords2() throws IOException {
        for(int j=0;j<kata.length;j++)
        {
            for(int i=0;i<dokumen.length;i++)
            {
                bagOfWords2[i][j] = tf[i][j]*idf[j];
            }
        }

        FileWriter write2 = new
FileWriter("D:/bagOfWords2.txt");
        PrintWriter print2_line = new PrintWriter(write2);
        print2_line.printf(("TF-IDF%n"));
        for(int i=0;i<dokumen.length;i++)
        {
            for(int j=0;j<kata.length;j++)
            {
                print2_line.printf("%s ",bagOfWords2[i][j]);
            }
            print2_line.printf("%n");
        }
        print2_line.close();
    }

    public List<String> getIdentitasKategori() {
        return identitasKategori;
    }

    public void setIdentitasKategori() {
        BagValuePosition bvp[] = new BagValuePosition[25];
        for(int i=0;i<25;i++)
        {
            BagValuePosition temp = new BagValuePosition();
            temp.setValue(Double.MIN_VALUE);
            temp.setPosKata(0);
        }
    }

```

```

        temp.setPosDokumen(0);
        bvp[i] = temp;
    }

    boolean cekKolom[] = new boolean[1000];
    for(int i=0;i<dokumen.length;i++)
    {
        for(int j=0;j<kata.length;j++)
        {
            if(bagOfWords2[i][j] > bvp[0].getValue())
            {
bvp[24].setValue(bvp[23].getValue());bvp[24].setPosDokumen(b
vp[23].getPosDokumen());bvp[24].setPosKata(bvp[23].getPosKat
a());

bvp[23].setValue(bvp[22].getValue());bvp[23].setPosDokumen(b
vp[22].getPosDokumen());bvp[23].setPosKata(bvp[22].getPosKat
a());
        . . .
            }
            else if(bagOfWords2[i][j] >
bvp[24].getValue())
            {
                bvp[24].setValue(bagOfWords2[i][j]);
                bvp[24].setPosDokumen(i);
                bvp[24].setPosKata(j);
                cekKolom[j] = true;
            }
        }
    }
    List<String> tempList = new ArrayList<String>();
    for(int i=0;i<25;i++)
    {
        String temp_kata = "";
        temp_kata =
kata[bvp[i].getPosKata()].toString();
        tempList.add(temp_kata);
        //System.out.println(temp_kata+"
"+bvp[i].getValue()+" "+bvp[i].getPosDokumen()+"
"+bvp[i].getPosKata()+" "+dokumen[bvp[i].getPosDokumen()]);
        //System.out.println(bvp[i].getValue());
        /*System.out.println(bvp[i].getPosDokumen());

System.out.println(dokumen[bvp[i].getPosDokumen()]);
        System.out.println(bvp[i].getPosKata());*/
    }

    for(int i=0;i<12;i++)

```

```

        {
            String temp = tambahID[id_kt-1][i];
            if(temp.equals(""))
                continue;
            else tempList.add(temp);
        }

        HashSet<String> set = new HashSet<String>(tempList);
        identitasKategori.addAll(set);
    }
}

```

### Kode Sumber 7.5 Kelas Training

```

public class Testing {

    private List<List<String>> identity;
    private int bagWords[][];
    private double bagWords2[][];
    private Object kata[];
    private int dokumen[];
    private double panjangdokumen[];
    private double cosSim[];
    private double tf[][];
    private double idf[];

    public Testing() {
        identity = new ArrayList<List<String>>();
        bagWords = new int[1000][1000];
        bagWords2 = new double[1000][1000];
        kata = new Object[5000];
        dokumen = new int[5];
        panjangdokumen = new double[6];
        cosSim = new double[6];
        tf = new double[1000][1000];
        idf = new double[1000];
    }

    public int[][] getBagWords() {
        return bagWords;
    }

    public void setBagWords() throws IOException {
        FileWriter write = new
FileWriter("D:/bagOfWordsTesting.txt");
        PrintWriter print_line = new PrintWriter(write);

        //perhitungan TF
        print_line.printf("%nTF%n");
    }
}

```

```

for(int i=0;i<dokumen.length;i++)
{
    //int max = 0;
    int totalwords = 0;
    List<String> lis2 = identity.get(i);
    for(int j=0;j<kata.length;j++)
    {
        String cari = kata[j].toString();
        int count = 0;
        for(int k=0;k<lis2.size();k++)
        {
            if(lis2.get(k).equalsIgnoreCase(cari))
                count++;
        }
        bagWords[i][j] = count;
        /*if(bagWords[i][j] > max)
            max = bagWords[i][j];*/
        totalwords += count;
    }

    for(int j=0;j<kata.length;j++)
    {
        tf[i][j]
(double)bagWords[i][j]/(double)totalwords;
        print_line.printf("%s ",tf[i][j]);
    }
    print_line.printf("%n");
}

//perhitungan DF
int totalPerTerm[] = new int[5000];
print_line.printf("%nDF%n");
for(int j=0;j<kata.length;j++)
{
    int jumlah = 0;
    for(int i=0;i<dokumen.length;i++)
    {
        jumlah += bagWords[i][j];
    }
    totalPerTerm[j] = jumlah;
    print_line.printf("%s ", totalPerTerm[j]);
}
print_line.printf("%n");

//perhitungan IDF
print_line.printf("%nIDF%n");
for(int j=0;j<kata.length;j++)
{

```

```

        idf[j] =
Math.log((double)dokumen.length/(double)totalPerTerm[j])/Mat
h.log(2);
        print_line.printf("%s ", idf[j]);
    }
    print_line.close();
}

public double[][] getBagWords2() {
    return bagWords2;
}

public void setBagWords2() throws IOException {
    for(int i=0;i<dokumen.length;i++)
    {
        double hitung = 0;
        for(int j=0;j<kata.length;j++)
        {
            bagWords2[i][j] = tf[i][j]*idf[j];
            double kuadrat = 0;
            kuadrat = bagWords2[i][j]*bagWords2[i][j];
            hitung += kuadrat;
        }
        panjangdokumen[i] = Math.sqrt((double)hitung);
    }

    FileWriter write2 = new
FileWriter("D:/bagOfWords2Testing.txt");
    PrintWriter print2_line = new PrintWriter(write2);
    print2_line.printf("%nTF-IDF%n");

    for(int i=0;i<dokumen.length;i++)
    {
        for(int j=0;j<kata.length;j++)
        {
            print2_line.printf("%s ",bagWords2[i][j]);
        }
        print2_line.printf("%n");
    }

    print2_line.printf("%nPanjang Dokumen %n");
    for(int k=0;k<panjangdokumen.length;k++)
        print2_line.printf("%s ",panjangdokumen[k]);
    print2_line.close();
}
}

```

### Kode Sumber 7.6 Kelas Testing



```

public int hitungCosineSimilarity(String stemmed, int
id_data, int act) throws IOException, SQLException
{
    double panjangDocQuery;
    String[] kataa = stemmed.split(",");

    List<Double> bagQuery = new ArrayList<Double>();
    int total = 0;
    for(int i=0;i<kata.length;i++)
    {
        String cek = kata[i].toString();
        int hitung = 0;
        for(int j=0;j<kataa.length;j++)
        {
            if(kataa[j].equalsIgnoreCase(cek))
                hitung++;
        }
        bagQuery.add((double)hitung);
        /*if(hitung > max)
            max = hitung;*/
        total += hitung;
    }

    List<Double> bagQuery2 = new ArrayList<Double>();
    double panjang = 0;
    for(int i=0;i<kata.length;i++)
    {
        double temp, tf_idf_q;
        temp = bagQuery.get(i);
        if(total == 0)
            temp = 0;
        else temp = temp/total;
        bagQuery.set(i, temp);
        tf_idf_q = temp*idf[i];
        bagQuery2.add(tf_idf_q);
        panjang += (tf_idf_q*tf_idf_q);
    }
    panjangDocQuery = Math.sqrt((double)panjang);
    FileWriter tulis = new
FileWriter("D:/bagOfWordsQueryTesting.txt", true);
    PrintWriter print_line = new PrintWriter(tulis);
    for(int i=0;i<bagQuery.size();i++)
        print_line.printf("%s ", bagQuery.get(i));
    print_line.printf("%n");
    print_line.close();

    FileWriter tulis2 = new
FileWriter("D:/bagOfWordsQuery2Testing.txt", true);
    PrintWriter print2_line = new PrintWriter(tulis2);

```

```

for(int i=0;i<bagQuery2.size();i++)
    print2_line.printf("%s ", bagQuery2.get(i));
print2_line.printf("\n");
print2_line.close();

for(int i=0;i<dokumen.length;i++)
{
    double atas = 0;
    double bawah = 0;
    double cosSimDoc = 0;
    for(int j=0;j<kata.length;j++)
    {
        double temp =
(double)bagWords2[i][j]*(double)bagQuery2.get(j);
        atas += temp;
    }
    bawah = panjangdokumen[i]*panjangDocQuery;
    if(panjangDocQuery==0)
    {
        for(int k=0;k<5;k++)
            cosSim[k] = 0.0;
        cosSim[5] = 1.000000;
    }
    else
    {
        cosSimDoc = atas/bawah;
        cosSimDoc =
Math.round(cosSimDoc*1000000.0)/1000000.0;
        cosSim[i] = cosSimDoc;
        cosSim[5] = 0.0;
    }
}

System.out.println("Panjang
dokumen="+panjangDocQuery);

MySQLAccess dao = new MySQLAccess();
dao.init();

if(act == 0)
{
    for(int i=0;i<cosSim.length;i++)
    {
        int j=i+1;
        dao.setMembershipValue(id_data, j,
cosSim[i]);
        System.out.println(j+" "+cosSim[i]);
    }
    int salah;
}

```

```

        salah = cekValidasiData(id_data);
        return salah;
    }
    else
    {
        for(int i=0;i<cosSim.length;i++)
        {
            int j=i+1;
            dao.setMembershipUji(id_data, j, cosSim[i]);
            System.out.println(j+" "+cosSim[i]);
        }
        return 0;
    }
}
}

```

### Kode Sumber 7.7 Fungsi Hitung Cosine Similarity pada Kelas Testing

```

<?php

if(!defined('BASEPATH'))
    exit('No direct script access allowed');

class Navigation extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        parse_str( $_SERVER['QUERY_STRING'],
$_REQUEST );
        $CI = & get_instance();
        $CI->config->load("facebook",TRUE);
        $config = $CI->config->item('facebook');
        $this->load->library('Facebook', $config);
    }

    function index()
    {
        $stat = $this->cekLogin();
        $data['cek'] = $stat['cek'];
        if($data['cek']=='FALSE')
        {
            $data['url'] = $stat['url'];
            $this->load->view('depan', $data);
        }
        else if($data['cek']=='TRUE')
        {
            $data['user'] = $stat['user'];
            $data['urlout'] = $stat['urlout'];
            $data['acc'] = $stat['acc'];
        }
    }
}

```

```

if(strcmp($data['user']['username'], "helena.n.pratiwi")==0)
    $this->load->view('home_adm', $data);
else
    $this->load->view('home', $data);
}

function cekLogin()
{
    $userId = $this->facebook->getUser();
    if($userId==0)
    {
        $data['cek'] = 'FALSE';
        $data['url'] = $this->facebook-
>getLoginUrl(array('scope'=>'email, user_birthday,
publish_stream, user_photos', 'photo_upload'));
    }
    else
    {
        $data['cek'] = 'TRUE';
        $data['user'] = $this->facebook-
>api('/me');
        $data['acc'] = $this->facebook-
>getAccessToken();
        $args['next'] =
site_url('navigation/logout');
        $data['urlout'] = $this->facebook-
>getLogoutUrl($args);
    }
    return $data;
}

function logout()
{
    $this->facebook->destroySession();
    //$this->session->sess_destroy(); //
Assuming you have session helper loaded
    $this->goToDepan();
}

function goToDepan()
{
    $stat = $this->cekLogin();
    $data['cek'] = $stat['cek'];
    if($data['cek']=='FALSE')
    {
        $data['url'] = $stat['url'];
        $this->load->view('depan', $data);
    }
}

```

```

        else if($data['cek']=='TRUE')
        {
            $data['user'] = $stat['user'];
            $data['urlout'] = $stat['urlout'];
            $data['acc'] = $stat['acc'];
            if(strcmp($data['user']['username'], "helena.n.pratiwi")==0)
                $this->load->view('home_adm', $data);
            else
                $this->load->view('home', $data);
        }

        function goToHalaman()
        {
            $id_kat = $this->input->post('kat');
            $it = $this->input->post('it');

            $stat = $this->cekLogin();
            $data['cek'] = $stat['cek'];
            $this->load->model('Post_model');
            $jml = count($this->Post_model-
>getPostByCat($id_kat));
            $jml_hlm = $jml / 15;
            if($jml%15 > 0)
                $jml_hlm = $jml_hlm + 1;

            $str = $this->Post_model-
>getPostCatLimit($id_kat,$it);
            $pos_tr = array();
            $pos = "";

            require('atur_posting.php');
            $ap = new Atur_posting();

            if($it > 0)
                $prev = $ap->setPrev($it,$id_kat);
            if((int)($jml/15) > $it)
                $lanj = $ap->setNext($it,$id_kat);
            if($data['cek']=='FALSE')
            {
                for($i=0;$i<count($str);$i++)
                    $pos .= $ap-
>showData($str[$i], 'N');
            }
            else if($data['cek']=='TRUE')
            {
                for($i=0;$i<count($str);$i++)
                    $pos .= $ap-
>showData($str[$i], 'Y');
            }
        }
    }
}

```

```

    }
    $pos .= "<div id='nav_paging'>
            <table border='0'
width='560px'><tr>
                                                    <td width='250px'
align='left'>";
        if(isset($prev))
            $pos .= $prev;
        $pos .= "</td><td width='60px'
align='center'>";
        $pos .= $ap->setJump($jml_hlm, $it, $id_kat);
        $pos .= "</td><td width='250px'
align='right'>";
        if(isset($lanj))
            $pos .= $lanj;
        $pos .= "</td></tr></table></div>";
        echo $pos;
    }

    function goToFashion()
    {
        $stat = $this->cekLogin();
        $data['cek'] = $stat['cek'];
        $this->load->model('Post_model');
        $jml = count($this->Post_model-
>getPostByCat('1'));
        $jml_hlm = $jml / 15;
        if($jml%15 > 0)
            $jml_hlm = $jml_hlm + 1;

        $tr = $this->Post_model-
>getPostCatLimit('1','0');
        $pos_tr = array();

        require('atur_posting.php');
        $ap = new Atur_posting();

        $data['lanj'] = "<div id='nav_paging'>

<table border='0' width='560px'>

<tr>

<td width='250px' align='left'></td>

<td width='60px' align='center'>";
        $data['lanj'] .= $ap-
>setJump($jml_hlm, '0', '1');
        $data['lanj'] .= "</td>

```

```

width='250px' align='right'>";
                                                                    <td
        if((int)($jml/15) > 0 && ($jml%15 > 0))
        {
            $data['lanj'] .= $ap-
>setNext('0','1');
        }
        $data['lanj'] .= "</td></tr></table>";
        $data['lanj'] .= "</div>";

        if($data['cek']=='FALSE')
        {
            $data['url'] = $stat['url'];
            for($i=0;$i<count($tr);$i++)
                $pos_tr[$i] = $ap-
>showData($tr[$i],'N');
            $data['data_tr'] = $pos_tr;

            $this->load->view('fashion', $data);
        }
        else if($data['cek']=='TRUE')
        {
            $data['user'] = $stat['user'];
            $data['urlout'] = $stat['urlout'];
            $data['acc'] = $stat['acc'];
            for($i=0;$i<count($tr);$i++)
                $pos_tr[$i] = $ap-
>showData($tr[$i],'Y');
            $data['data_tr'] = $pos_tr;
            if(strcmp($data['user']['username'], "helena.n.pratiwi")==0)
                $this->load->view('fashion_adm', $data);
            else
                $this->load->view('fashion_mem', $data);
        }

        . . .

/*end of file navigation.php*/
/*location: ./application/controllers/navigation.php*/

```

### Kode Sumber 7.8 Kelas Navigation

```

function cariPost()
{
    $kata = (string)$this->input->post('kata', true);
    $words = explode(' ', $kata);
    $result = array();

```

```

foreach($words as $param)
{
    $this->load->model('Post_model');
    $tr = $this->Post_model->findPost($param);
    $result = array_merge($result, $tr);
}
$pos_tr = array();

require('navigation.php');
$nav = new Navigation();
$data = $nav->cekLogin();

require('atur_posting.php');
$ap = new Atur_posting();
if($data['cek']=='FALSE')
{
    for($i=0;$i<count($result);$i++)
        $pos_tr[$i] = $ap->showData($result[$i], 'N');
    $data['res_tr'] = $pos_tr;
    $this->load->view('cari_barang', $data);
}
else if($data['cek']=='TRUE')
{
    for($i=0;$i<count($result);$i++)
        $pos_tr[$i] = $ap->showData($result[$i], 'Y');
    $data['res_tr'] = $pos_tr;
if(strcmp($data['user']['username'], "helena.n.pratiwi")==0)
    $this->load->view('cari_barang_adm', $data);
else
    $this->load->view('cari_barang_mem', $data);
}
}

```

**Kode Sumber 7.9 Fungsi CariPost pada Kelas Cari**

```

function postIklan()
{
    $konten = (string)$this->input->post('konten', true);
    $konten = strip_tags($konten);
    $link = (string)$this->input->post('link', true);

    if(isset($konten))
    {
        try
        {
            if(!empty($_FILES['gmb']['name']))
            {

```



```

        $config['upload_path'] = './upload/'; /* NB! create
this dir! */
        $config['allowed_types'] = 'gif|jpg|png|bmp|jpeg';
        $config['max_size']     = '500';
        $config['max_width']    = '2560';
        $config['max_height']   = '1600';
        $this->load->library('upload', $config);
        $upload = $this->upload->do_upload('gmb');
        if(!$upload)
            echo $this->upload->display_errors();

        $target_folder = './upload/';
        $file_path = $target_folder . $_FILES['gmb']['name'];

        $this->facebook->setFileUploadSupport(true);
        $postImage = $this->facebook-
>api("/232903756901321/photos", 'post', array('source'=>
'@'.realpath($file_path),
                                     'message'=> $konten,
                                     'link'=> $link));
        }
        else
        {
            $postGroup = $this->facebook-
>api("/232903756901321/feed", 'post', array('message'=>
$konten,
                                     'link'=> $link));
        }
        $data['pesan'] = '<h3>Sukses Memasang
Iklan</h3>';

        require('navigation.php');
        $nav = new Navigation();
        $stat = $nav->cekLogin();
        $data['cek'] = $stat['cek'];

        if($data['cek']=='FALSE')
        {
            $data['url'] = $stat['url'];
            $this->load->view('depan');
        }
        else if($data['cek']=='TRUE')
        {
            $data['user'] = $stat['user'];
            $data['urlout'] = $stat['urlout'];
            $data['acc'] = $stat['acc'];
            if(strcmp($data['user']['username'], "helena.n.pratiwi
") == 0)
                $this->load->view('pasang_iklan_adm', $data);

```

```

        else
            $this->load->view('pasang_iklan', $data);
        }
    }
    catch(FacebookApiException $e)
    {
        //error_log($e);
        echo $e;
    }
}
}

```

**Kode Sumber 7.10 Fungsi PostIklan pada Kelas Write\_wall**

```

function like()
{
    require('navigation.php');
    $nav = new Navigation();
    $data = $nav->cekLogin();
    if($data['cek']=='TRUE')
    {
        $id_post = $this->input->post('ids');
        $acc = $data['acc'];
        try
        {
            $this->facebook-
>api("/$id_post/likes", 'post', array('access_token' =>
$acc));
            echo 'Sukses memberi like!';
            //echo $id_post;
        }
        catch(FacebookApiException $e)
        {
            //error_log($e);
            echo $e;
        }
    }
}
}

```

**Kode Sumber 7.11 Fungsi Like pada Kelas Mem\_only**

```

function comment()
{
    require('navigation.php');
    $nav = new Navigation();
    $data = $nav->cekLogin();
    if($data['cek']=='TRUE')
    {
        $konten = $this->input->post('konten');
    }
}

```

```

        $id_post = $this->input->post('ids');
        $sacc = $data['acc'];
        if(isset($konten))
        {
            try
            {
                $this->facebook-
>api("/$id_post/comments", 'post', array('message' =>
$konten,
                'access_token' => $acc));
                echo 'Sukses memberi
komentar!';
            }
            catch(FacebookApiException $e)
            {
                echo $e;
            }
        }
    }
}

```

**Kode Sumber 7.12 Fungsi Comment pada Kelas Mem\_only**

```

function hapus()
{
    $id_data = $this->input->post('ids');
    $kategori = $this->input->post('kats');
    $this->load->model('Post_model');
    $hps = $this->Post_model-
>updateMembership($id_data,$kategori);
    if($hps==1)
        echo 'Sukses menghapus posting dari
kategori';
    else echo $hps;
}

```

**Kode Sumber 7.13 Fungsi Hapus pada Kelas Admin\_only**

```

function setThres()
{
    $checked = $this->input->post('opt');
    $this->load->model('Post_model');
    $this->Post_model->hapusIsIncluded();
    if($checked==1)
    {
        $metode = $this->input->post('ddlMetode');
        if($metode==1 || $metode==2)
            $a = $this->Post_model-
>setIsIncluded($metode);
    }
}

```

```

        else
            $a = $this->Post_model-
>setAverage($metode);
            if($a==1)
                $data['pesan'] = '<h3>Sukses mengatur
nilai threshold</h3>';
            else $data['pesan'] = '<h3>Gagal,
'.'.a.'</h3>';
        }
        else
        {
            $value = $this->input->post('txtTres');
            $edit = $this->Post_model-
>updateThreshold($value);
            if($edit==1)
                $data['pesan'] = '<h3>Sukses mengatur
nilai threshold</h3>';
            else $data['pesan'] = '<h3>Gagal,
'.'.a.'</h3>';
        }
        require('navigation.php');
        $nav = new Navigation();
        $stat = $nav->cekLogin();
        $data['cek'] = $stat['cek'];
        $data['user'] = $stat['user'];
        $data['urlout'] = $stat['urlout'];
        $data['acc'] = $stat['acc'];

        if($data['cek']=='TRUE' &&
strcmp($data['user']['username'], "helena.n.pratiwi")==0)
        {
            $this->load->view('dashboard2', $data);
        }
    }
}

```

**Kode Sumber 7.14 Fungsi SetThres pada Kelas Admin\_only**

## LAMPIRAN B. HASIL PENGUJIAN

**Tabel 8.1 Pengujian Fitur Melihat Data Posting Skenario 2**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario 2</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kategori makanan.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori makanan ditunjukkan pada Gambar 8.1. Contoh data posting kategori makanan ditunjukkan pada Gambar 8.2.



**Gambar 8.1 Hasil Uji Fitur Melihat Data Posting Kategori Makanan**

MAU MENIKMATI MINUMAN SUSU COKELAT SEPERTI DI RESTO-RESTO? gak perlu lagi pergi ke resto, cukup bikin sendiri dan bisa dinikmati di rumah/kos sambil mengerjakan tugas.

KINI TELAH HADIR

Produk Nestle professional merupakan produk yang biasa dipakai di beberapa restoran terkenal, seperti KFC, Hoka-hoka Bento, McD, Pizza Hut, Indolicious Cafe Resto dan sebagainya.

RASA DIJAMIN ENAK dan SAMA seperti yang ada di restoran!

Serta memiliki citarasa yang berbeda, tidak dijual dipasaran, memiliki label halal bersertifikat dari MUI, dan terdaftar di BPOM, Kandungan zat bergizinya sangat baik untuk kesehatan.

milko: 75.000

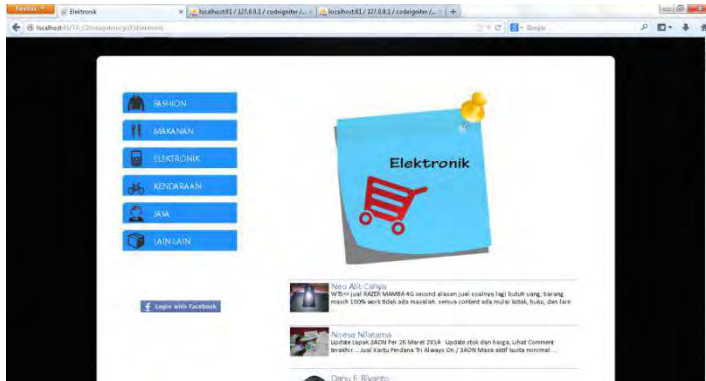
ovaltine: 70.000

MINAT, BISA PM atau SMS 089676000971

**Gambar 8.2 Contoh Posting Kategori Makanan**

**Tabel 8.2 Pengujian Fitur Melihat Data Posting Skenario 3**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario 3</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kategori elektronik.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori elektronik ditunjukkan pada Gambar 8.3. Contoh posting kategori elektronik ditunjukkan pada Gambar 8.4.



**Gambar 8.3 Hasil Uji Fitur Melihat Data Posting Kategori Elektronik**

(WTS) HP Sony Xperia J. Spesifikasi bisa dibrowsing. Masih mulus, dipakai sudah 1th.

Gratis Case lumayan baru...Berminat langsung PM ya. Alasan jual karena mau beli yang baru.

**Gambar 8.4 Contoh Posting Kategori Elektronik**

**Tabel 8.3 Pengujian Fitur Melihat Data Posting Skenario 4**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario 4</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.



<b>Data Uji</b>	Data uji berupa kategori kendaraan.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori kendaraan ditunjukkan pada Gambar 8.5. Contoh posting kategori kendaraan ditunjukkan pada Gambar 8.6.



**Gambar 8.5 Hasil Uji Fitur Melihat Data Posting Kategori Kendaraan**

Jual vespa apa adanya  
 no mesing ma body tembus  
 surat lengkap  
 pgn lbh jelasnya leat d fb ma k tkp lgs cod its  
 hub 089677333066/081330066622

**Gambar 8.6 Contoh Posting Kategori Kendaraan**

**Tabel 8.4 Pengujian Fitur Melihat Data Posting Skenario 5**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario 5</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kategori jasa.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil

<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori jasa ditunjukkan pada Gambar 8.7. Contoh posting kategori kendaraan ditunjukkan pada Gambar 8.8.
----------------------	---



**Gambar 8.7 Hasil Uji Fitur Melihat Data Posting Kategori Jasa**

ngapain repot2 berdesak-desakan d stasiun kalau bisa pesan tiket lewat HP?

cara pemesanan tiket:

1. sebutkan tanggal dan rute
2. nanti kami akan sebutkan kereta dan kereta yang tersedia berikut harga tiketnya
3. sebutkan nama dan no identitas diri/KTP calon penumpang serta email
4. transfer sejumlah harga tiket yang dipesan segera setelah pemesanan (max 2 jam)
5. konfirmasi transfernya
6. anda akan menerima email pemesanan tiket, print dan tukarkan d stasiun sebelum anda berangkat

CALL/SMS/WA : 081325888599

happy travelling ^^

**Gambar 8.8 Contoh Posting Kategori Jasa**

**Tabel 8.5 Pengujian Fitur Melihat Data Posting Skenario 6**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur melihat data posting dari aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur untuk menampilkan posting yang termasuk dalam kategori yang dipilih oleh pengguna.
<b>Skenario 6</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan ketika pengguna memilih kategori tertentu.
<b>Kondisi Awal</b>	Halaman dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kategori lain-lain.
<b>Langkah Pengujian</b>	Pengguna memilih salah satu kategori yang disediakan pada halaman web aplikasi.
<b>Hasil Yang Diharapkan</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Yang Didapat</b>	Data posting yang termasuk dalam kategori terpilih ditampilkan pada halaman web aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan halaman data posting kategori lain-lain ditunjukkan pada Gambar 8.9. Contoh posting kategori lain-lain ditunjukkan pada Gambar 8.10.



**Gambar 8.9 Hasil Uji Fitur Melihat Data Posting Kategori Lain-lain**

Kertas Murah meriah gan untuk keperluan skripsi dan bikin laporan.  
 mulai dari 60-120 gram gan.  
 bisa beli perbox atau perrim.  
 berbagai merk dari yang paling murah hingga kualitas mumpuni...,  
 Gratis antar daerah ITS dan sekitarnya.  
 hubungi ane : 083857612234  
 Matur Nuwun.,

**Gambar 8.10 Contoh Posting Kategori Lain-lain**

**Tabel 8.6 Pengujian Fitur Mencari Posting Tertentu Skenario 2**

<b>ID</b>	UJ.UC-0002
<b>Referensi Kasus Penggunaan</b>	UC-0002
<b>Nama</b>	Pengujian fitur mencari posting tertentu pada aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur mencari posting berdasarkan kata kunci dari pengguna.
<b>Skenario</b>	Posting yang bersesuaian dengan kata kunci dari pengguna akan ditampilkan pada panel hasil pencarian.

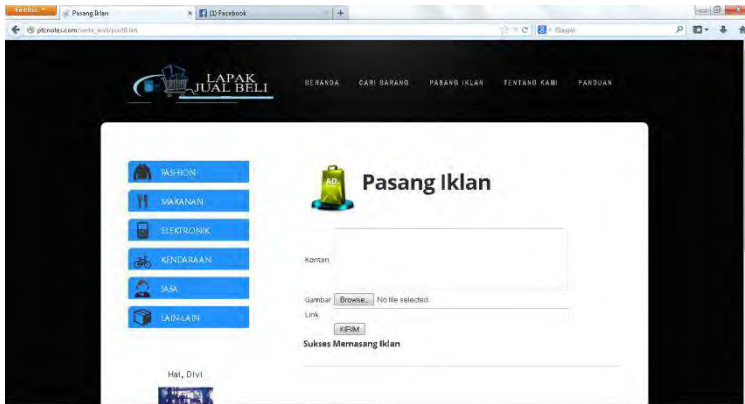
<b>Kondisi Awal</b>	Halaman Cari Barang dari aplikasi telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa kata kunci pencarian yaitu 'dinosaurus'.
<b>Langkah Pengujian</b>	Pengguna memasukkan kata kunci pencarian dan memilih tombol 'cari'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan notifikasi bahwa data posting tidak ditemukan.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan notifikasi bahwa data posting tidak ditemukan.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan aplikasi menampilkan notifikasi hasil pencarian ditunjukkan pada Gambar 8.11.



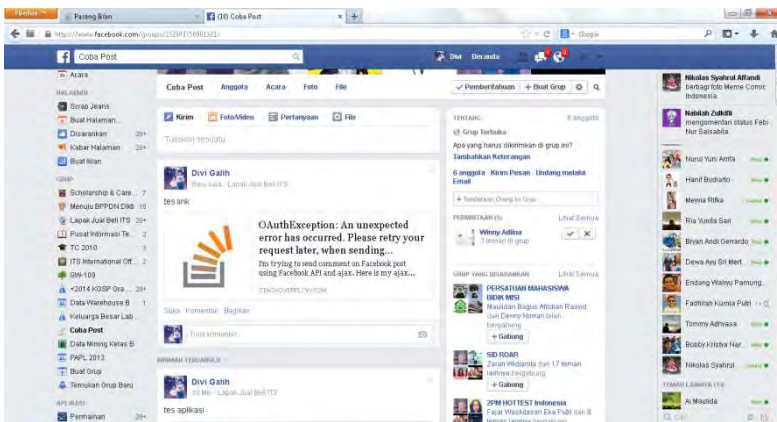
**Gambar 8.11 Hasil Uji Fitur Mencari Posting Skenario 2**

**Tabel 8.7 Pengujian Fitur Mem-posting Iklan Skenario 2**

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur mem-posting iklan melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur menulis pada wall grup melalui aplikasi.
<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi dan posting akan ditampilkan pada wall grup.
<b>Kondisi Awal</b>	Pengguna merupakan anggota grup dan telah login dengan akun Facebook. Halaman Pasang Iklan telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa konten teks 'Tes link' dan sebuah link sebagai lampiran.
<b>Langkah Pengujian</b>	Pengguna memasukkan konten iklan dan link sebagai lampiran lalu memilih tombol 'kirim'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting baru ditampilkan pada wall grup uji coba.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem pada halaman web aplikasi dan data posting baru ditampilkan pada wall grup uji coba.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan respon berhasil dari sistem ditunjukkan pada Gambar 8.12. Tampilan data posting baru pada wall grup Facebook ditunjukkan pada Gambar 8.13.



Gambar 8.12 Hasil Uji Fitur Mem-posting Iklan Skenario 2



Gambar 8.13 Tampilan Posting Baru pada Wall Grup

Tabel 8.8 Pengujian Fitur Mem-posting Iklan Skenario 3

<b>ID</b>	UJ.UC-0003
<b>Referensi Kasus Penggunaan</b>	UC-0003
<b>Nama</b>	Pengujian fitur mem-posting iklan melalui aplikasi.

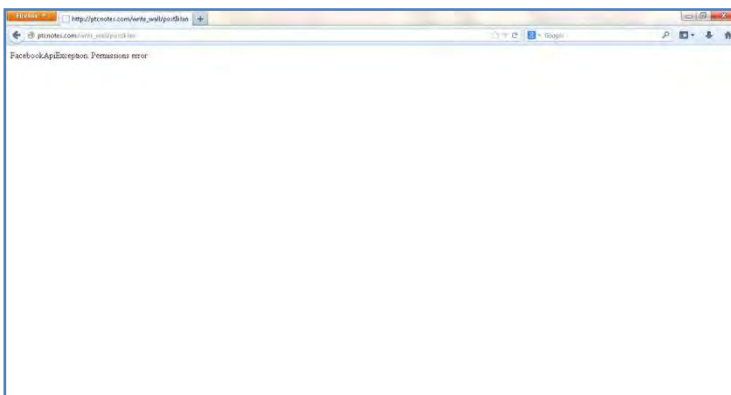


<b>Tujuan Pengujian</b>	Menguji fitur pengecekan anggota ketika hendak menulis pada wall grup pada aplikasi.
<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi.
<b>Kondisi Awal</b>	Pengguna bukan merupakan anggota grup dan telah login dengan akun Facebook. Halaman Pasang Iklan telah dibuka pada browser.
<b>Data Uji</b>	Data uji berupa konten teks 'Tes aplikasi'.
<b>Langkah Pengujian</b>	Pengguna memasukkan konten iklan lalu memilih tombol 'kirim'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan notifikasi error pada halaman aplikasi.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan notifikasi error pada halaman aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan respon gagal dari sistem ditunjukkan pada Gambar 8.14.

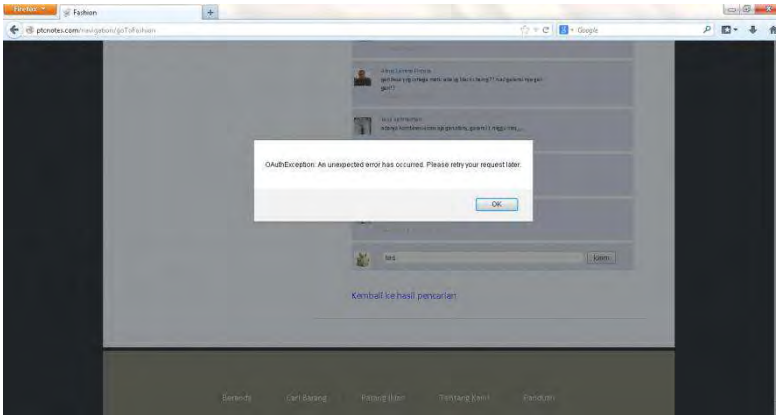
**Tabel 8.9 Pengujian Fitur Menulis Komentar di Posting Skenario 2**

<b>ID</b>	UJ.UC-0004
<b>Referensi Kasus Penggunaan</b>	UC-0004
<b>Nama</b>	Pengujian fitur menulis komentar pada posting melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur pengecekan anggota ketika hendak menulis komentar pada posting.

<b>Skenario</b>	Respon dari sistem akan ditampilkan pada halaman aplikasi.
<b>Kondisi Awal</b>	Pengguna bukan merupakan anggota dari grup dan telah login dengan akun Facebook. Pengguna telah memilih posting yang ingin diberi komentar.
<b>Data Uji</b>	Data uji berupa konten komentar yang ingin diberikan dan posting yang ingin diberi komentar.
<b>Langkah Pengujian</b>	Pengguna memasukkan konten komentar pada textbox yang tersedia dan memilih tombol 'kirim'.
<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan notifikasi error pada halaman aplikasi.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan notifikasi error pada halaman aplikasi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan notifikasi error dari sistem ditunjukkan pada Gambar 8.15.



**Gambar 8.14 Tampilan Notifikasi Error Mem-posting Iklan**



**Gambar 8.15 Tampilan Notifikasi Error Menulis Komentar pada Posting**

**Tabel 8.10 Pengujian Fitur Memberi Like pada Posting Skenario 2**

<b>ID</b>	UJ.UC-0005
<b>Referensi Kasus Penggunaan</b>	UC-0005
<b>Nama</b>	Pengujian fitur memberikan like pada posting melalui aplikasi.
<b>Tujuan Pengujian</b>	Menguji fitur pengecekan anggota ketika hendak memberi like pada posting.
<b>Skenario</b>	Penanda bahwa posting telah diberikan like ditampilkan pada tab like dari posting.
<b>Kondisi Awal</b>	Pengguna bukan merupakan anggota dari grup dan telah login dengan akun Facebook. Pengguna telah memilih posting yang ingin diberi like.
<b>Data Uji</b>	Data uji berupa posting yang ingin diberi like.
<b>Langkah Pengujian</b>	Pengguna memilih pilihan 'Suka' pada posting.

<b>Hasil Yang Diharapkan</b>	Aplikasi menampilkan respon berhasil dari sistem dan menampilkan tulisan 'Disukai' pada tab like dari posting.
<b>Hasil Yang Didapat</b>	Aplikasi menampilkan respon berhasil dari sistem dan menampilkan tulisan 'Disukai' pada tab like dari posting.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Tampilan tab like dari posting terpilih ditunjukkan pada Gambar 8.17.

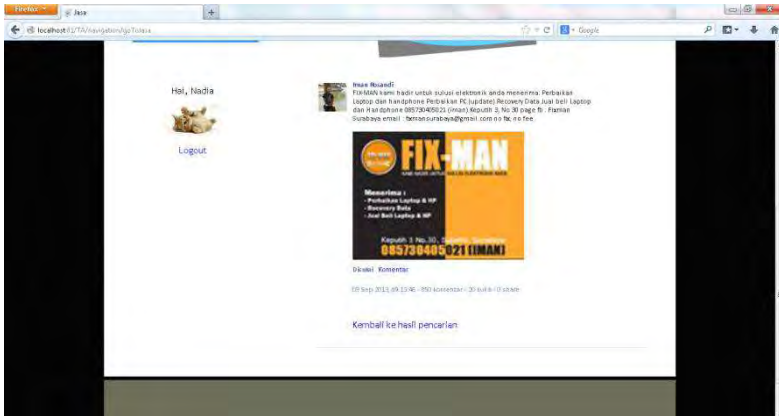
**Tabel 8.11 Profil Responden Pengguna Aplikasi**

Responden	Jenis Kelamin	Usia (Tahun)	Pekerjaan
1	Laki-laki	21	Mahasiswa
2	Laki-laki	21	Mahasiswa
3	Perempuan	20	Pegawai
4	Perempuan	22	Mahasiswa
5	Laki-laki	22	Mahasiswa
6	Perempuan	21	Mahasiswa
7	Laki-laki	21	Mahasiswa
8	Perempuan	22	Mahasiswa
9	Perempuan	21	-
10	Perempuan	22	Mahasiswa
11	Laki-laki	21	Mahasiswa
12	Laki-laki	21	Mahasiswa
13	Laki-laki	21	Mahasiswa
14	Perempuan	20	-
15	Perempuan	22	Mahasiswa
16	Laki-laki	21	Mahasiswa
17	Perempuan	21	Mahasiswa
18	Laki-laki	21	Mahasiswa
19	Perempuan	21	-
20	Laki-laki	21	Mahasiswa

**SURVEI PENDAPAT PENGGUNA APLIKASI**

1. Apakah Anda sering merasa kesulitan dalam mencari informasi tertentu pada timeline grup di Facebook?
  - a. Ya
  - b. Tidak
2. Apakah proses pengelompokan posting grup berdasarkan kategori pada aplikasi ini mempermudah Anda dalam mencari informasi tertentu?
  - a. Ya
  - b. Tidak
3. Bagaimana pendapat Anda mengenai tingkat akurasi hasil pengelompokan posting pada aplikasi ini?
  - a. Sangat baik
  - b. Baik
  - c. Cukup
  - d. Kurang
  - e. Sangat kurang
4. Bagaimana pendapat Anda mengenai tingkat responsivitas aplikasi ini?
  - a. Baik
  - b. Cukup
  - c. Kurang
5. Bagaimana pendapat Anda mengenai fitur-fitur yang terdapat pada aplikasi ini? Apakah mempermudah pengguna dalam mengirim atau menerima informasi ke/dari grup?
  - a. Baik
  - b. Cukup
  - c. Kurang
6. Jika kurang, menurut Anda fitur apa yang seharusnya tersedia atau diperbaiki dari aplikasi?

**Gambar 8.16 Kuisioner Pengujian Kegunaan Aplikasi**



**Gambar 8.17 Hasil Uji Fitur Memberi Like pada Posting Skenario 2**

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

#### **6.1. Kesimpulan**

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Aplikasi dapat melakukan pengelompokan data posting grup Facebook dengan menggunakan metode klasifikasi Fuzzy dan pembobotan TF-IDF dengan tingkat ketepatan sekitar 0,76. Berdasarkan hasil pengujian, hasil terbaik diperoleh jika nilai threshold Fuzzy di-set sebesar 0,1.
2. Aplikasi dapat mempermudah pengguna dalam mencari informasi jual beli barang. Hasil survei pada 20 orang pengguna aplikasi dimana 95% pengguna menyatakan aplikasi dapat membantu pengguna dalam melakukan pencarian informasi.
3. Pengaturan nilai threshold Fuzzy memiliki pengaruh terhadap tingkat ketepatan dari aplikasi. Berdasarkan hasil pengujian, nilai threshold Fuzzy yang optimal berada pada kisaran 0,1 hingga 0,125.

#### **6.2. Saran**

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Di antaranya adalah sebagai berikut:

1. Meningkatkan tingkat ketelitian dari sistem dengan menambah jumlah data pada kategori sehingga model yang dihasilkan memiliki kualitas yang lebih baik.

2. Menambahkan fitur pemberian tagging untuk mempermudah proses pencarian informasi.
3. Menambahkan analisis tambahan untuk perbaikan pengelompokan posting yang memiliki lebih dari satu kategori.



## BIODATA PENULIS



Penulis, Helena Nadia Pratiwi lahir di Malang pada 11 Januari 1992. Penulis adalah anak kedua dari tiga bersaudara dan dibesarkan di Malang, Jawa Timur.

Penulis menempuh pendidikan formal di MIN Malang I (1998-2004), SMPN 3 Malang (2004-2007), dan SMAN 3 Malang (2007-2010). Pada tahun 2010, penulis memulai pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak dan memiliki ketertarikan di bidang SQA, basis data, software evolution, dan software maintenance. Penulis juga aktif dalam organisasi kemahasiswaan yaitu Himpunan Mahasiswa Teknik Computer (HMTTC). Penulis sempat menjadi asisten dosen selama menjalani perkuliahan di Teknik Informatika yaitu asisten dosen mata kuliah Data Mining. Penulis dapat dihubungi melalui alamat email [helenanadia.pratiwi@gmail.com](mailto:helenanadia.pratiwi@gmail.com).