

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis

Analisis merupakan tahapan penting dalam menyusun suatu sistem, dimana tujuannya adalah untuk merancang sistem dengan baik, mengidentifikasi serta mengatasi permasalahan yang rumit, sehingga solusi yang ditemukan dapat diimplementasikan secara terstruktur dan dapat dipahami dengan mudah. Integrasi analisis dalam pengembangan sistem menjadi kunci bagi penyusunan yang teratur dan efisien.

4.4.1 Analisis Permasalahan

Penyewa jasa menghadapi tantangan serius akibat kurangnya naungan yang memadai dalam menyelenggarakan acara. Hal ini dapat mengakibatkan masalah seperti kesulitan dalam mencari jasa serta kurangnya naungan. Salah satu solusi yang dapat diadopsi adalah dengan mengimplementasikan e-service ini, seperti yang telah dijelaskan sebelumnya. Dengan menggunakan platform e-service yang menggabungkan elemen shopping search engine dan metode TOPSIS berdasarkan review dan harga, Penyewa dapat dengan lebih efisien mencari dan memilih penyedia layanan yang diperlukan dalam penyewaan jasa. Penerapan e-service akan memungkinkan event organizer untuk menjelajahi berbagai opsi penyedia layanan, termasuk paket harga, rekomendasi produk dengan lebih terstruktur dan efektif. Data yang dikumpulkan melalui proses crawling akan memberikan informasi lebih lanjut dalam penjelajahan semisal produk yang di cari tidak ditemukan di e-service, dan membantu memberikan data untuk pengambilan keputusan yang lebih baik. Selain itu, penggunaan metode *TOPSIS* akan membantu penyewa jasa dalam mengevaluasi berbagai alternatif penyedia layanan berdasarkan faktor review dan harga. Hal ini akan membantu mereka dalam memilih opsi yang paling sesuai dengan kebutuhan acara, sambil mempertimbangkan kualitas pelayanan dan anggaran yang tersedia. Dengan demikian, integrasi e-service dalam proses penyewaan jasa dan acara dapat mengatasi kendala yang diakibatkan oleh kurangnya naungan. Ini akan membantu penyewa untuk lebih efisien, mendapatkan layanan yang tepat, serta meningkatkan kualitas acara yang dihasilkan.

4.4.2 Analisis Kebutuhan sistem

Analisis kebutuhan sistem digunakan untuk mengetahui kebutuhan apa saja yang akan digunakan dalam proses perancangan sistem. Pada bagian ini akan diuraikan dengan jelas kebutuhan sistem yang akan dibuat. Analisa ini dilakukan pada bagian *shopping search engine* sebagai kebutuhan yang perlu dipenuhi dalam pembuatan sistem. Dengan analisa ini, menjalankan sistem akan dapat lebih maksimal jika dibandingkan dengan menjalankan sistem tanpa melakukan analisa terlebih dahulu. Dalam sub bab ini, akan menjelaskan langkah-langkah yang diambil untuk melakukan pengumpulan kebutuhan yang diperlukan dalam pengembangan crawler data dari website <https://www.sejasa.com/jasa/semua-layanan>. Pengumpulan kebutuhan ini bertujuan untuk merancang dan mengembangkan sistem crawler yang efektif dan efisien dalam mengambil data dari website tersebut. Berdasarkan hasil analisa terhadap kebutuhan dalam perancangan sistem yang sedang dikembangkan dalam pengembangan ini, dibutuhkan *shopping search engine* sebagai berikut:

1. Identifikasi Kebutuhan Data

Pertama-tama, melakukan identifikasi kebutuhan data yang relevan dengan skripsi ini. Selanjutnya menentukan variabel-variabel data yang perlu diekstrak dari halaman website tersebut. Informasi yang akan diambil meliputi aspek-aspek seperti nama produk, website, rating, url, gambar dan category. Identifikasi ini akan membantu merumuskan target data yang akan diambil oleh sistem crawler.

2. Analisis Struktur Halaman

Setelah kebutuhan data diidentifikasi, selanjutnya akan melakukan analisis mendalam terhadap struktur HTML halaman website. Tujuannya adalah untuk memahami bagaimana data disajikan dalam kode HTML. Dengan memahami struktur ini, kami akan dapat mengidentifikasi elemen-elemen yang perlu diekstrak, memeriksa penggunaan tag HTML, kelas, dan ID dalam halaman untuk menentukan bagaimana data dapat diambil dengan akurat. Gambar 4.1 menampilkan contoh kode yang mengilustrasikan urutan pengambilan struktur halaman HTML, seperti yang dijelaskan pada penjelasan di bawah ini:

```

// Mengambil nama produk
$productElement = $xpath->query( "//div[@class='profile-area_desc']/a[@class='profile-area_desc--name'] , $productElement->Item());
if ($productElement) {
    $nama_produk = $productElement->InnerText();
    $productData['nama_produk'] = $nama_produk;
} else {
    echo "Nama produk tidak ditemukan.";
}

```

Gambar 4.1 Mengambil data nama produk

Pada gambar 4.1 Kode itu menggunakan PHP dan XPath untuk mengambil teks dari elemen HTML. Kode Itu akan mencari elemen “h2” dengan kelas “profile-area__desc--name” di dalam elemen div dengan kelas profile-area__desc, yang terletak dalam elemen yang disebut “\$productElement”. Jika elemen ditemukan, teks di dalamnya diambil dan disimpan dalam variable” \$productData['nama_produk']”. Jika tidak ditemukan, akan dicetak pesan "Nama produk tidak ditemukan."

Langkah berikutnya adalah mengekstrak data yang mencakup informasi mengenai nama website sebagai bagian dari proses pengambilan data. Selain itu, pada Gambar 4.2 terdapat potongan kode yang menjelaskan implementasi langkah-langkah pengambilan data, termasuk proses ekstraksi informasi nama website, yang diterapkan dalam skenario pengembangan.

```

// Mengambil data nama website
$websiteElement = $xpath->query( '//a[@class="navbar-new__brand"]/img' ]->item(0);
if ($websiteElement) {
    $websiteAlt = $websiteElement->getAttribute('alt');
    $websiteName = substr($websiteAlt, 0, strpos($websiteAlt, '.com') + 4);
    $productData['website'] = $websiteName;
}

```

Gambar 4.2 Mengambil data nama website

Pada gambar 4.2 Kode tersebut menggunakan PHP dan XPath untuk mengambil nama website dari atribut "alt" suatu gambar di dalam elemen "a" dengan kelas "navbar-new__brand". Jika elemen ditemukan, kode ini mengambil nilai atribut "alt", memprosesnya untuk mendapatkan nama website (hingga ".com"), dan menyimpannya dalam variabel “\$productData['website']”.

Langkah ketiga dalam proses ini adalah mengambil data rating dari sumber informasi yang relevan untuk melengkapi dataset. Pada Gambar 4.3, terdapat potongan kode yang menggambarkan implementasi langkah-langkah pengambilan data rating, memberikan pemahaman lebih mendalam tentang bagaimana proses ini dijalankan dalam konteks pengembangan aplikasi.

```

$reviewElement = $xpath->query('.//div[@class="profile-area__desc--review"]', $productElement)->Item(0);
if ($reviewElement) {
    $ratingValue = $reviewElement->textContent;
    $productData['rating'] = $ratingValue;
} else {
    echo "Data tidak ditemukan.";
}

```

Gambar 4.3 Mengambil data rating

Pada gambar 4.3 Kode tersebut menggunakan PHP dan XPath untuk mengambil informasi ulasan dari elemen HTML. Ia mencari elemen “div” dengan kelas “profile-area__desc—review” di dalam elemen “\$productElement”. Jika elemen ditemukan, kode ini mengambil teks di dalamnya (nilai peringkat ulasan) dan menyimpannya dalam variabel “\$productData['rating']”. Jika elemen tidak ditemukan, akan dicetak pesan "Data tidak ditemukan."

Langkah keempat melibatkan pengambilan data URL sebagai bagian integral dari proses pengumpulan informasi. Pada Gambar 4.4, terdapat potongan kode yang memperinci implementasi langkah-langkah pengambilan data URL, memberikan gambaran yang jelas tentang bagaimana proses ini diintegrasikan dalam pengembangan aplikasi.

```

$productData['url_asli'] = $url;

```

Gambar 4.4 Mengambil data Url

Pada Gambar 4.4 Baris kode tersebut bertujuan untuk mengassign nilai dari variabel \$url ke elemen khusus dengan kunci 'url_asli' dalam array \$productData. Dengan kata lain, variabel atau array \$productData digunakan untuk menyimpan informasi produk, dan 'url_asli' menjadi kunci yang menunjukkan atribut URL asli dari produk tersebut. Proses ini memungkinkan pengelolaan dan penyimpanan data URL asli produk dalam struktur data yang terorganisir.

Langkah kelima melibatkan pengambilan data atribut tambahan sebagai bagian integral dari proses pengumpulan informasi. Pada Gambar 4.5, terdapat potongan kode yang menjelaskan implementasi langkah-langkah pengambilan data atribut tambahan, memberikan pemahaman lebih mendalam tentang bagaimana proses ini diintegrasikan dalam pengembangan aplikasi.

```

// Mengambil URL gambar
$imgElement = $xpath->query('//div[@class="profile-area_picture"]/img')->item(0);
if ($imgElement) {
    $imgSrc = $imgElement->getAttribute('src');
    $productData['gambar_url'] = $imgSrc;
} else {
    echo "Data tidak ditemukan.";
}

```

Gambar 4.5 Mengambil data gambar

Pada gambar 4.5 Kode ini menggunakan PHP dan XPath untuk mengambil URL gambar dari atribut "src" dalam elemen "img", yang terletak di dalam elemen "div" dengan kelas "profile-area_picture". Jika elemen gambar ditemukan, URL-nya diambil dan disimpan dalam variabel "\$productData['gambar_url']". Jika tidak ditemukan, akan dicetak pesan "Data tidak ditemukan."

Langkah terakhir dalam proses ini melibatkan penyimpanan hasil akhir dari pengambilan data ke dalam struktur variabel atau array. Pada Gambar 4.6, keterangan menunjukkan bahwa nilai yang dienkripsi dari pencarian produk disimpan dalam variabel \$productData['category']. Tahapan ini menyelesaikan proses pengumpulan informasi dengan 'category' sebagai kunci yang menyimpan informasi terenkripsi mengenai kategori produk dalam struktur data akhir.

```

// Menyimpan kategori ke dalam daftar
$productData['category'] = $search_product_encoded;

```

Gambar 4.6 Mengambil data category

Pada gambar 4.6 Kode ini, memiliki nilai yang dienkripsi dari pencarian produk disimpan dalam variabel "\$productData['category']".

3. Penanganan Halaman Dinamis

Apabila website menggunakan teknologi pemuatan konten dinamis melalui JavaScript dan Paginasi, kami akan mempertimbangkan penggunaan library DOMXPath dan DomDocument seperti untuk mengambil data dari elemen yang dimuat secara dinamis. Penggunaan teknologi semacam ini akan memungkinkan sistem crawler untuk mengakses data yang dihasilkan secara dinamis dan pengambilan data di paginasi oleh website. Contoh kode yang dirancang untuk menghasilkan antarmuka pengguna sebagaimana diilustrasikan

pada Gambar 4.7. Pada gambar tersebut, terlihat hasil tangkapan layar seperti berikut:

```
$baseUrl = 'https://www.sejasa.com/mitra-kami/';
$search_product_encoded = str_replace(' ', '-', $search_product);
$search_product_encoded = urlencode($search_product_encoded);
$currentPage = 1;
$maxPage = 2; // Jumlah halaman yang ingin diambil "Jumlah": Unknown word.
$productDataList = [];
```

Gambar 4. 7 Inisiasi Halaman Paginasi di Crawler

Pada gambar 4.7 kode ini menetapkan variabel dan pengaturan awal untuk melakukan penelusuran halaman web. Ini meliputi URL dasar (`baseUrl`) situs web yang akan dijelajahi, kata kunci pencarian yang telah diubah bentuknya untuk URL (`search_product_encoded`), nomor halaman saat ini (`currentPage`), dan jumlah maksimal halaman yang akan diambil datanya (`maxPage`). Data produk yang ditemukan akan disimpan dalam “productDataList”.

Langkah selanjutnya melibatkan proses pengambilan URL halaman konten sebagai bagian integral dari pengembangan aplikasi. Pada Gambar 4.8, terdapat penjelasan dan implementasi terinci mengenai langkah-langkah yang diambil untuk mengambil URL halaman konten. Informasi ini memberikan pemahaman mendalam tentang strategi pengelolaan dan penggunaan URL dalam konteks pengembangan aplikasi ini.

```
$url = $baseUrl . $search_product_encoded . '?page=' . $currentPage;
$file = @file_get_contents($url);
```

Gambar 4. 8 pengambilan url halaman content

Pada gambar 4.8 Kode tersebut menggabungkan elemen-elemen untuk membentuk URL lengkap dengan kata kunci pencarian dan nomor halaman. Kemudian, ia menggunakan `file_get_contents` untuk mengambil konten dari halaman web yang diakses melalui URL tersebut.

4. Penanganan Error dan Retry

Kami akan merancang mekanisme untuk mengatasi berbagai jenis error yang mungkin terjadi selama proses crawling. Mekanisme retry akan diterapkan untuk mengatasi koneksi terputus, halaman tidak ditemukan, atau halaman yang tidak merespons. Dengan demikian, sistem kami akan dapat mengatasi kendala teknis dengan baik. Dapat dilihat pada potongan gambar 4.9:

```

if ($file === false) {
    $error_message = "Sorry, the product you're looking for is not available.";
    return view('front.products.not_found')->with(compact('error_message'));
}

```

Gambar 4. 9 Penanganan error retry

5. Penyimpanan dan Pengelolaan Data

Setelah data berhasil diekstrak, kami akan mempertimbangkan opsi penyimpanan yang tepat untuk data yang diambil. Pilihan meliputi penyimpanan dalam basis data. Penyimpanan yang terstruktur akan memudahkan akses dan manajemen data.

6. Label produk

Berdasarkan proses algoritma TOPSIS yang diimplementasikan, produk diberi label berdasarkan peringkat relatif mereka dalam kaitannya dengan kriteria harga. Proses ini melibatkan analisis harga untuk produk-produk yang kemudian dinormalisasi, dibobotkan, dan dinilai relatif satu sama lain. Produk kemudian diberi label atau identifikasi berdasarkan urutan mereka dari yang memiliki skor relatif tertinggi ke terendah, menandai produk-produk tersebut sebagai "produk unggulan" atau "produk rekomendasi" dalam konteks perbandingan relatif terhadap kriteria harga yang diterapkan. Berikut kode program untuk algoritma TOPSIS:

```

// Hitung nilai maksimum dan minimum harga
$maxHarga = $categoryproduct->max('harga');
$minHarga = $categoryproduct->min('harga');

// Normalisasi harga untuk setiap produk
$categoryproduct->transform(function ($product) use ($maxHarga, $minHarga) {
    $product->normalized_harga = ($product->harga - $minHarga) / ($maxHarga - $minHarga);
    return $product;
});

// Hitung bobot kriteria (hanya harga)
$weightHarga = 1; // Karena hanya ada satu kriteria
$categoryproduct->transform(function ($product) use ($weightHarga) {
    $product->weighted_harga = $product->normalized_harga * $weightHarga;
    return $product;
});

// Hitung skor relatif
$relativeScores = [];
foreach ($categoryproduct as $product) {
    $relativeScores[$product->id] = $product->weighted_harga;
}

// Urutkan produk berdasarkan skor relatif (TOPSIS)
arsort($relativeScores);

// Ambil urutan ID produk berdasarkan skor relatif
$sortedProductIds = array_keys($relativeScores);

// Ambil data produk yang diurutkan sesuai dengan urutan skor relatif
$sortedProducts = $categoryproduct->whereIn('id', $sortedProductIds)->values();
// dd($sortedProducts);

```

Gambar 4. 10 Label produk menggunakan TOPSIS

4.4.3 Kebutuhan Fungsional

Dalam upaya meningkatkan pengalaman pengguna, sistem e-service ini dirancang dengan memperhatikan sejumlah kebutuhan fungsional yang krusial. Kami akan menjelajahi beberapa fitur utama yang telah diimplementasikan dalam sistem ini, yang bertujuan untuk memberikan pelayanan yang lebih efektif dan memuaskan bagi pengguna. Kebutuhan fungsional dalam sistem e-service ini mencakup sejumlah fitur penting yang dirancang untuk meningkatkan pengalaman pengguna. Beberapa fitur utama yang disediakan oleh sistem ini dijelaskan secara rinci pada Tabel 4.1. Tabel ini memberikan gambaran menyeluruh tentang berbagai fitur yang terintegrasi dalam sistem, termasuk fungsionalitas, opsi, dan kegunaan masing-masing fitur untuk mendukung pengalaman pengguna secara optimal.:

Tabel 4. 1 Kebutuhan Fungsional

No	Fitur	Tentang Fitur
1.	Shopping Search Engine	Penyewa dapat dengan mudah mencari berbagai layanan yang mereka butuhkan untuk penyewaan jasa, seperti label rekomendasi dan product unggulan, menggunakan mesin pencari khusus. Sistem mengaplikasikan metode TOPSIS untuk menganalisis berbagai alternatif penyedia layanan berdasarkan faktor review dan harga, memberikan rekomendasi yang sesuai dengan preferensi penyewa.
2.	Penilaian kualitas layanan	Penyewa dapat melihat penilaian dan ulasan dari penyedia layanan sebelum mereka mengambil keputusan, membantu dalam mengevaluasi kualitas pelayanan yang diharapkan.
3.	Cart	Saat penyewa menjelajahi produk atau layanan yang ditawarkan di e-service, mereka memiliki opsi untuk menambahkan item yang menarik perhatian mereka ke dalam keranjang belanja. Ini bisa berupa berbagai jenis produk, layanan, atau komponen acara dalam sistem e-service. Setiap kali pengguna menambahkan item, informasi terkait seperti nama produk, gambar, harga, dan jumlah yang diinginkan akan disimpan dalam keranjang.

4.	Checkout	Fitur checkout pada sistem ini memungkinkan penyewa untuk menambahkan beberapa alamat pengiriman yang berbeda sesuai dengan kebutuhan mereka, dengan opsi pembayaran hanya melalui metode Cash on Delivery (COD).
----	----------	---

4.4.4 Kebutuhan Non-Fungsional

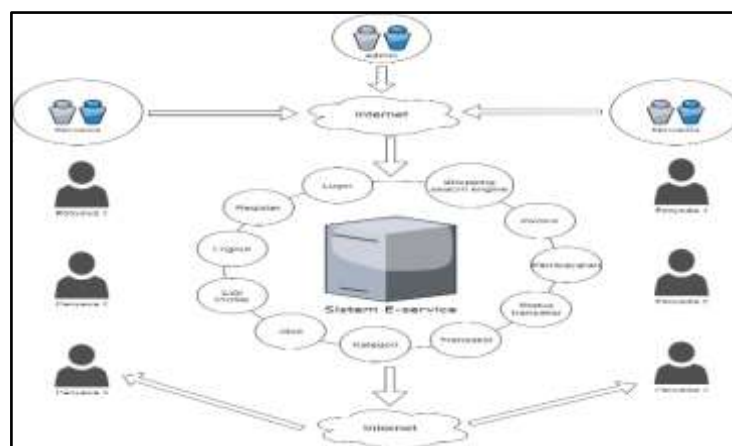
Kebutuhan non-fungsional adalah spesifikasi atau persyaratan sistem yang berkaitan dengan kemampuan yang disediakan dalam pengembangan ini. Beberapa kebutuhan yang ada dalam sistem ini antara lain:

1. Sistem ini memiliki batasan penggunaan di mana ia hanya dapat diakses pada server yang terhubung ke jaringan lokal atau intranet. Kendala ini muncul karena aplikasi berbentuk web dan saat ini hanya dapat diakses melalui localhost, karena masih dalam tahap pengembangan sebelum di-hosting secara publik.

4.2 Perancangan Sistem

4.2.1 Arsitektur sistem

Arsitektur sistem adalah struktur komponen-komponen dalam suatu sistem teknologi informasi. Ini melibatkan desain perangkat lunak, perangkat keras, dan jaringan serta cara komponen-komponen tersebut saling berinteraksi. Arsitektur sistem membantu komponen sistem beroperasi efisien sesuai dengan tujuan yang ditetapkan. Penjelasan lebih lanjut dapat dilihat di Gambar 4.11.



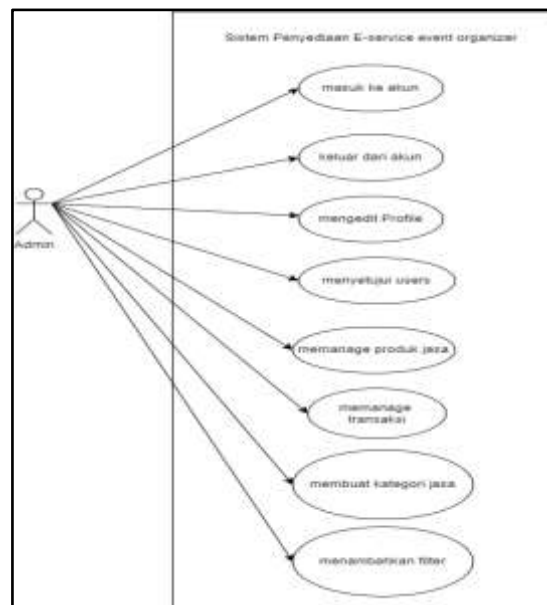
Gambar 4. 11 Arsitektur sistem

Pada gambar 4.11 menjelaskan tentang bagaimana alur sistem diakses dan fitur apa saja yang ada dalam sistem.

Pada gambar 4.12 menjelaskan hubungan antar class dari website *e-service* dapat dilihat bahwasanya. Dalam konteks sistem yang melibatkan tiga peran pengguna utama, yaitu admin, penyedia, dan penyewa, class diagram adalah representasi visual dari struktur dan hubungan kelas-kelas yang terlibat dalam sistem perangkat lunak. Pada sistem ini, Penyewa memiliki kemampuan untuk melakukan banyak pencarian, namun data yang terkait hanya disimpan sekali dalam pencarian. Selain itu, Penyewa dapat melakukan banyak pesanan yang kemudian diikuti oleh proses transaksi, pembayaran, dan penerbitan invoice untuk setiap pesanan yang dilakukan. Pengguna juga memiliki kemampuan untuk membuat banyak review terhadap produk yang disediakan oleh penyedia. Di sisi penyedia, mereka dapat membuat banyak kategori dan menyediakan beragam jasa, serta menerima ulasan untuk setiap layanan yang mereka tawarkan.

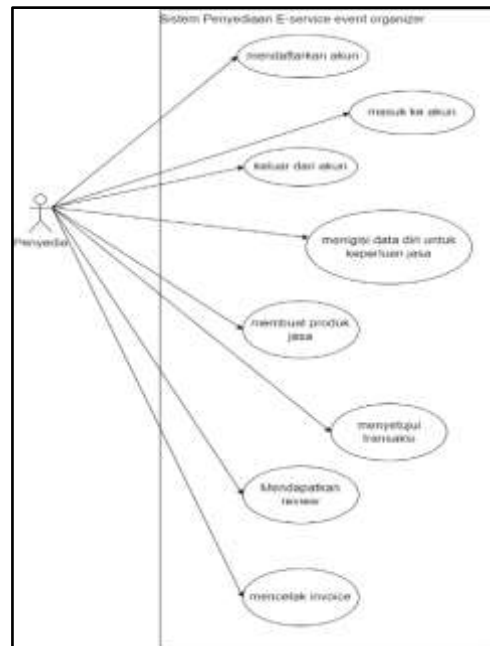
4.2.3 Usecase

Rancangan sistem penyediaan *e-service* untuk event organizer melibatkan tiga jenis pengguna dengan peran yang berbeda, yaitu Admin, Penyedia, dan Penyewa. Untuk menggambarkan interaksi dan fungsi yang relevan dengan peran masing-masing, telah disusun diagram use case. Diagram use case Gambar 4.13 menggambarkan fungsionalitas Admin dalam sistem *e-service* untuk event organizer, termasuk pengelolaan event, informasi penyedia layanan, dan kontrol umum atas operasional sistem.



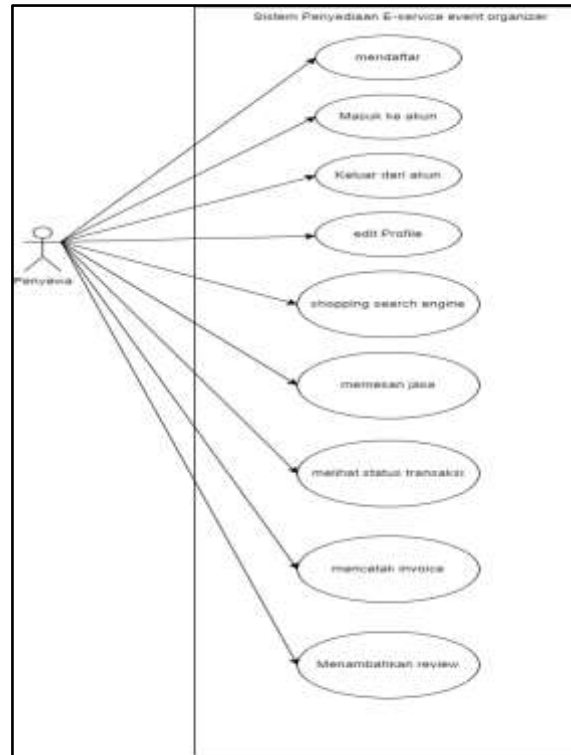
Gambar 4. 13 Usecase Admin

Gambar 4.14 menyajikan diagram use case yang mengilustrasikan peran dan interaksi pengguna dengan peran Penyedia dalam sistem e-service. Diagram ini memberikan gambaran holistik mengenai fungsionalitas dan akses yang dimiliki oleh Penyedia, mencakup proses manajemen dan penyediaan layanan e-service sesuai dengan kebutuhan acara yang diorganisir.



Gambar 4. 14 Usecase Penyedia

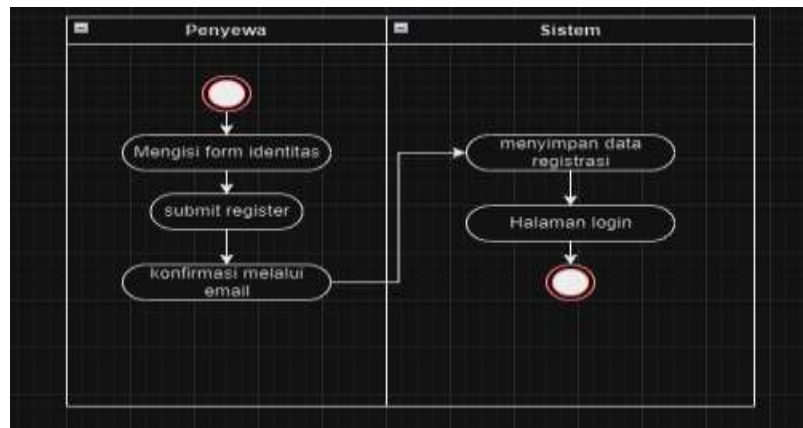
Sementara itu, Gambar 4.15 menampilkan diagram use case untuk pengguna dengan peran Penyewa. Pada diagram ini, terlihat interaksi dan fungsionalitas yang dapat diakses oleh Penyewa dalam kaitannya dengan pemanfaatan layanan e-service yang disediakan oleh penyedia



Gambar 4. 15 Usecase Penyewa

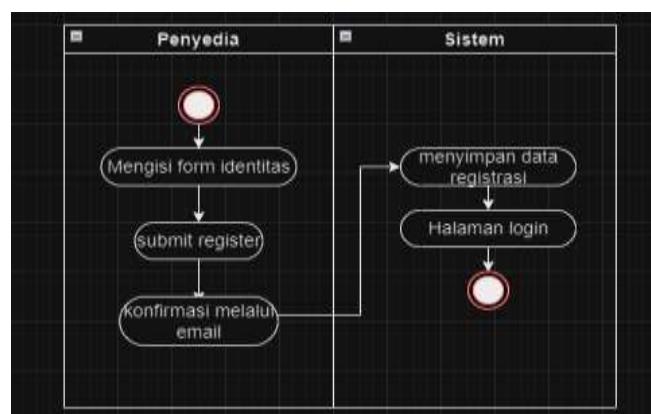
4.2.4 Activity diagram

Activity diagram adalah bentuk visual yang memuat aktivitas dan tindakan, juga dikenal sebagai diagram aktivitas. Diagram ini mampu menggambarkan pilihan, pengulangan, dan konkurensi. Penggunaannya meliputi penjelasan aktivitas di dalam sistem komputer maupun alur kegiatan di dalam sebuah organisasi. Gambar 4.16 menampilkan diagram aktivitas yang merinci langkah-langkah proses pendaftaran penyewa dalam sistem. Diagram ini memberikan gambaran visual mengenai urutan kegiatan dan interaksi antara penyewa dengan sistem pada saat proses pendaftaran. Aktivitas-aktivitas tersebut mencakup langkah-langkah seperti pengisian formulir, verifikasi identitas, dan penyelesaian proses pendaftaran secara umum.



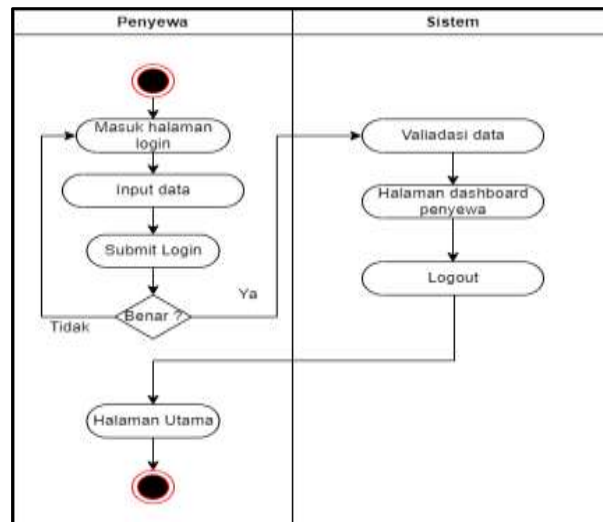
Gambar 4. 16 Activity diagram register penyewa

Gambar 4.17 memperlihatkan diagram aktivitas yang menggambarkan secara rinci proses pendaftaran penyedia dalam sistem. Diagram ini memberikan visualisasi langkah-langkah interaktif yang dilakukan oleh penyedia dalam mendaftarkan diri ke dalam sistem, mencakup kegiatan seperti pengisian formulir, verifikasi informasi, dan penyelesaian proses pendaftaran secara keseluruhan.



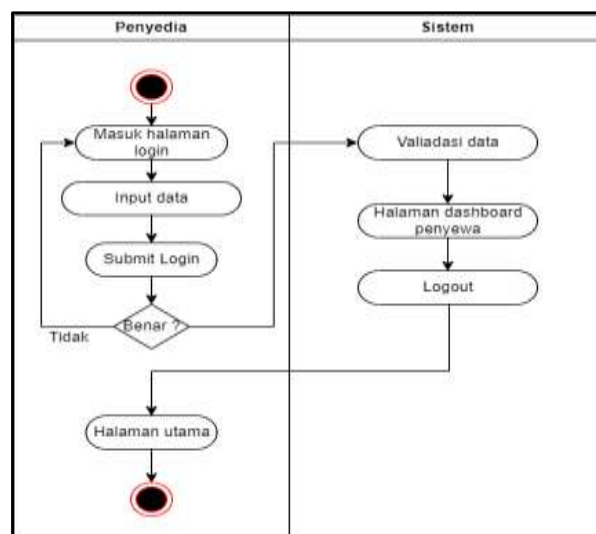
Gambar 4. 17 Activity Diagram register penyedia

Gambar 4.18 menggambarkan diagram aktivitas yang menunjukkan serangkaian langkah yang diperlukan ketika seorang penyewa melakukan proses login dalam sistem. Diagram ini memberikan visualisasi terperinci tentang interaksi antara penyewa dan sistem pada tahap login, mencakup aktivitas seperti penginputan kredensial, verifikasi identitas, dan akses ke fungsi-fungsi tertentu setelah login berhasil.



Gambar 4. 18 Activity diagram penyewa login

Gambar 4.19 memperlihatkan diagram aktivitas yang merinci serangkaian langkah dalam proses login penyedia dalam sistem. Diagram ini memberikan gambaran visual terperinci mengenai interaksi antara penyedia dan sistem pada saat login, mencakup aktivitas seperti penginputan kredensial, verifikasi identitas, dan akses ke fungsi-fungsi tertentu setelah login berhasil.

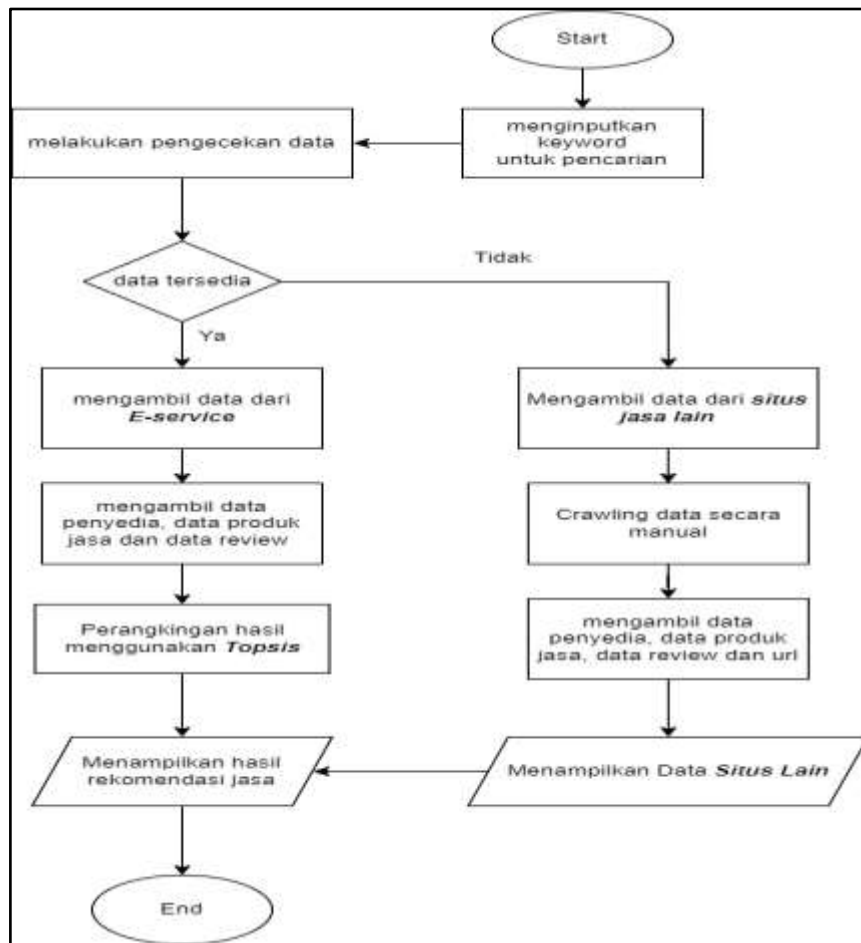


Gambar 4. 19 Activity diagram penyedia login

4.2.5 Flowcart diagram

Aplikasi e-service ini menggunakan fitur utama berupa shopping search engine. Shopping Search Engine adalah platform online untuk mencari dan membandingkan produk. Membantu pengguna menemukan informasi harga,

ulasan, dan penawaran produk secara cepat dan efisien. Alur dari shopping search engine ini direpresentasikan dalam bentuk flowchart pada gambar 4.20:



Gambar 4. 20 Flowcart shopping search engine

Pada gambar 4.20 Proses pencarian pada aplikasi ini dimulai dari halaman utama, di mana pengguna memasukkan kata kunci (keyword). Sistem kemudian mengambil data dari e-service. Data yang diambil meliputi informasi penyedia, kategori, ulasan, dan harga (yang diambil secara manual). Jika tidak ada data yang sesuai yang ditemukan, website ini akan memuat data dari situs lain. Data yang terkumpul, seperti kelompok harga dan kelompok kategori, akan digunakan sebagai kriteria dalam pengambilan keputusan menggunakan metode TOPSIS. Produk yang dipilih akan diidentifikasi sebagai produk unggulan atau produk rekomendasi.