
Perancangan Lokapasar Terdistribusi Sebagai Publikasi Kalatog Produk UMKM

Haddad Sammir¹, Khairil Hamdi²

hsammir@jayanusa.ac.id¹, Khairilhamdi@ jayanusa.ac.id²

^{1,2}STMIK Jaya Nusa

Informasi Artikel

Diterima : 06-08-2025

Direview : 20-08-2025

Disetujui : 31-08-2025

Kata Kunci

Lokapasar, UMKM,
Aplikasi Terdistribusi,
Socket.io

Abstrak

Pengunaan teknolgi informasi dalam penjualan adalah hal yang tak terhindarkan. Ada banyak pendekatan yang dapat digunakan seperti mengembangkan toko online atau bergabung dengan lokapasar populer. Usaha Mikro, Kecil dan Menengah (UMKM) dapat menggunakan metoda di atas, namun masih mendapatkan hambatan seperti kurangnya literasi literasi digital serta kebutuhan sumber daya dan biaya yang tinggi. Mengembangkan lokapasar untuk mendukung UMKM juga memiliki tantangan tersendiri seperti kebutuhan teknis dan biaya yang tinggi. Oleh karena itu, penelitian ini menawarkan solusi yang mengambil jalan tengah dengan membangun lokapasar terdistribusi dengan membagi beban teknis lokapasar kepada pengembang dan pelaku UMKM dengan tetap mengedepankan efisiensi dan kemdahan pengoperasian.

Keywords

Marketplace, MSME,
Distributed Application,
Socket.io

]

Abstrak

The use of information technology in sales is inevitable. There are many approaches that can be used such as developing an online store or joining a popular marketplace. Micro, Small and Medium Enterprises (MSMEs) can use the above methods, but still encounter obstacles such as lack of digital literacy and high resource and cost requirements. Developing a marketplace to support MSMEs also has its own challenges such as high technical requirements and costs. Therefore, this study offers a solution that takes a middle ground by building a distributed marketplace by sharing the technical burden of the marketplace with developers and MSME actors while still prioritizing efficiency and ease of operation.

A. Pendahuluan

Sebagai salah satu sector usaha yang tahan krisis dan sebagai katalisator perekonomian, Usaha Mikro, Kecil dan Menengah (UMKM) perlu diperkuat dengan digitalisasi dan pemanfaatan teknologi informasi. Hal tersebut menjadi penting karena pemanfaatan teknologi informasi mampu meningkatkan transformasi bisnis, ketepatan dan efisiensi pertukaran informasi yang berdampak pada terciptanya UMKM berdaya saing [1].

Bentuk penerapan teknologi informasi pada UMKM diantaranya adalah pemanfaatan lokapasar dan e-katalog. Secara umum penelitian yang membahas lokapasar mengarahkan pelaku UMKM untuk memasarkan produknya pada lokapasar yang umum digunakan. Dua diantara lokapasar yang paling banyak digunakan adalah Tokopedia dan Shopee dan setiap tahun terjadi peningkatan penggunaan marketplace oleh pengguna internet [2].

Sementara itu penelitian yang membahas e-katalog secara umum membahas pembuatan pamphlet digital yang dapat disematkan pada halaman web atau chanel digital lainnya. Penelitian Prameswari et al memfokuskan pembuatan katalog digital dalam format pdf yang dipublikasikan secara online pada website heyzine [3]. Penelitian selanjutnya yang dilakukan oleh Rustriani et al membuat katalog dalam format cetak yang menjangkau kalangan yang lebih tua serta katalog dalam format digital yang menjangkau masyarakat yang lebih muda [4].

Dari penjabaran lokapasar dan katalog di atas dapat dilihat bahwa para peneliti masih memfokuskan katalog yang khusus dirancang untuk satu UMKM. Belum terdapat integrasi katalog antar UMKM yang dapat memperluas pencarian sebuah produk. Penelitian ini menggabungkan lokapasar dengan katalog elektronik yang mampu melakukan pencarian katalog kepada setiap UMKM yang mempublikasikan katalognya sehingga pencarian sebuah produk dapat dilakukan lintas UMKM.

B. Tinjauan Pustaka

2.1 UMKM

Usaha mikro, kecil dan menengah (UMKM) adalah kegiatan usaha yang dijalankan oleh individu, rumah tangga, atau badan usaha berskala kecil yang digolongkan berdasarkan pendapatan per tahun, jumlah karyawan dan kepemilikan asset [5]. Berdasarkan kepemilikan asset, usaha kecil memiliki kekayaan bersih lebih dari Rp 50.000.000 (lima puluh juta rupiah) sampai paling banyak Rp 500.000.000 (lima ratus juta rupiah) dengan penjualan tahunan lebih dari Rp 300.000.000 (tiga ratus juta rupiah) sampai paling banyak Rp 2.500.000.000 (dua milyar lima ratus juta rupiah). Usaha menengah memiliki kekayaan bersih lebih dari Rp 500.000.000 (lima ratus juta rupiah) sampai paling banyak Rp 10.000.000.000 (sepuluh milyar rupiah) dan lebih dari Rp 2.500.000.000 (dua milyar lima ratus juta rupiah) sampai paling banyak Rp 50.000.000.000 (lima puluh milyar rupiah) [6].

2.2 Lokapasar

Lokapasar atau yang lebih umum dikenal sebagai marketplace adalah sistem informasi yang memungkinkan penjual dan pembeli mengkomunikasikan produk, harga dan melakukan transaksi melalui jalur elektronik [7]. Selain itu, lokapasar juga dapat diartikan sebagai sebuah pasar elektronik yang melakukan kegiatan menjual dan membeli suatu barang ataupun jasa yang meliputi tiga aspek

yaitu business to business (b2), business to customer (b2c) dan customer to customer (2c) yang didominasi oleh B2B sebesar 75% [8]. Lokapasar menjadi penting bagi UMKM karena lokapasar dapat meningkatkan jumlah khalayak dan penjualan UMKM serta memberinya akses ke layanan logistik dan proses pembayaran [2].

2.3 Katalog

Katalog merupakan media pemasaran untuk memperkenalkan produk kepada pelanggan dengan memberikan informasi lengkap berupa tampilan produl, harga dan informasi lainnya yang relevan [4]. Katalog dalam bentuk elektronik atau E-Katalog dapat membantu UMKM memperluas jangkauan pasar dan meningkatkan daya saing karena e-katalog dapat didistribusikan dalam berbagai kanal elektronik yang tersedia [9].

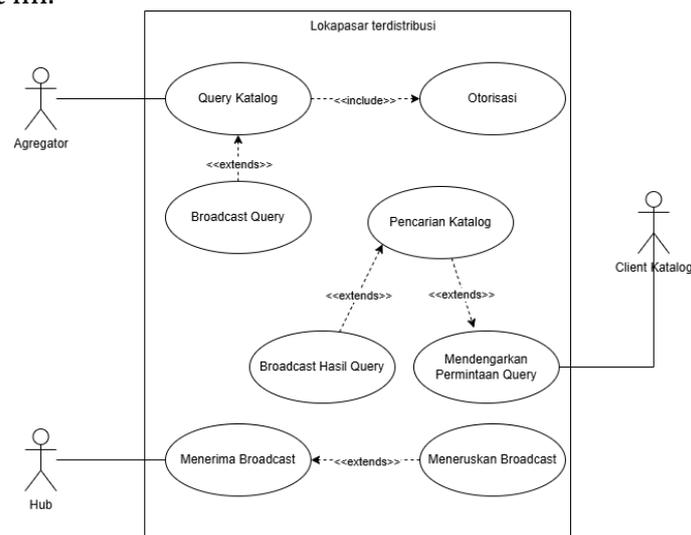
2.4 Socket.io

Socket.io adalah pustaka javascript *client-server* yang memungkinkan terjadinya komunikasi dua arah secara *real time* antara pengguna. Socket.io banyak digunakan pada aplikasi yang membutuhkan komunikasi real time seperti aplikasi Emergency Response [10], aplikasi online game fighter plane yang diteliti oleh Pratama dan Arisna [11], serta aplikasi chat yang dikembangkan oleh Pandiri, et al [12].

B. METODE PENELITIAN

Lokapasar terdistribusi terdiri dari beberapa komponen yaitu aggregator yang berfungsi sebagai pengumpul informasi dari berbagai sumber, klien katalog yang menyediakan informasi serta sebuah hub yang memiliki peran sebagai penghubung. Penelitian ini memanfaatkan pustaka socketio sebagai pondasi komunikasi antar komponen. Implementasi yang digunakan adalah berbasis web, meskipun dapat menggunakan implementasi lain seperti desktop sesuai dengan implementasi yang disediakan oleh socketio.

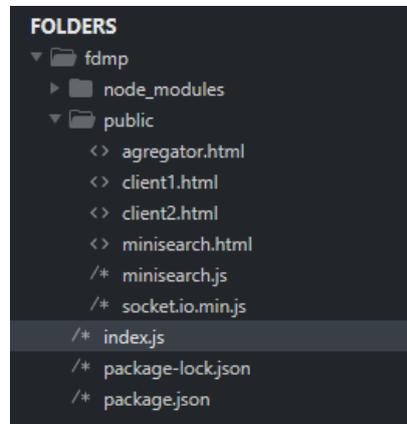
Aktor, operasi dan interaksi antar komponen dapat dilihat pada diagram use case berikut ini.



Gambar 1. Use Case Diagram

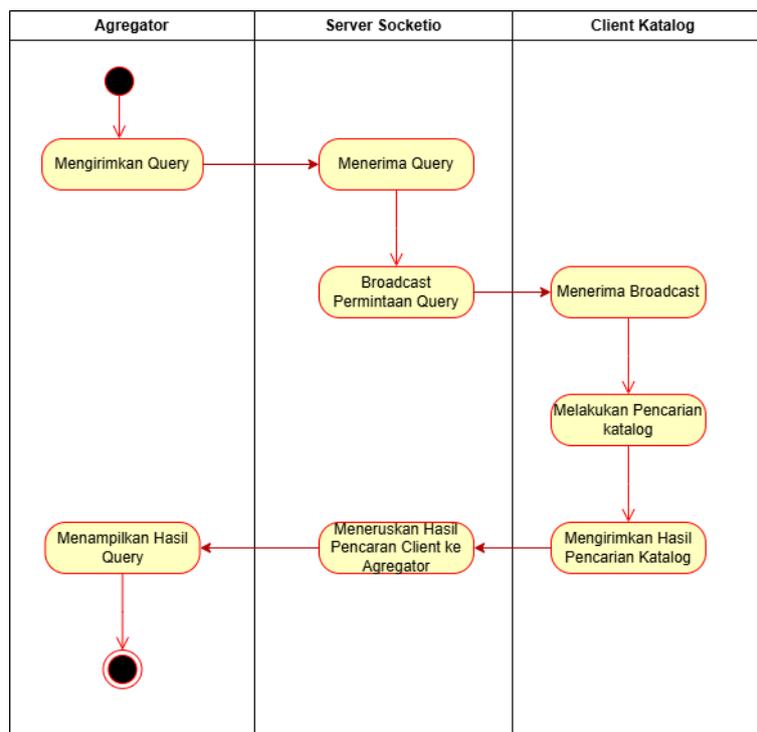
Penelitian ini terdiri dari sebuah application server nodejs yang sudah terinstall dan melibatkan pustaka socketio dan expressjs. Pustaka ini dapat diinstall menggunakan node package manager (npm) dan nantinya akan

ditempatkan pada direktori “node_modules”. Komponen server penelitian ini ditulis pada file index.js, sementara komponen klien ditulis didalam skrip javascript yang dikombinasikan pada file html. Struktur direktori aplikasi dapat dilihat pada gambar di bawah ini.



Gambar 2. Struktur Direktori Aplikasi

Sistem yang dirancang membutuhkan sebuah aggregator dan setidaknya satu client penyedia katalog. Alur kerja komponen sistem ditampilkan pada gambar 3 berikut ini.



Gambar 3. Activity Diagram

Activity diagram di atas menggambarkan alur komunikasi antara aggregator sebagai pengumpul informasi katalog dengan client yang menyediakan katalog. Permintaan query yang bersumber dari pencarian katalog oleh pengguna dikirimkan ke hub socketio server untuk di-broadcast ke semua client katalog yang terhubung. Client yang menerima broadcast akan melakukan pencarian pada

katalognya dan mengirimkan kembali hasil pencarian ke hub. Hub selanjutnya meneruskan semua respon hasil pencarian ke aggregator yang akan menampilkannya ke pengguna.

Persiapan Sistem

Penelitian ini secara umum menggunakan nodejs sebagai application server beserta modul express sebagai server web dan socketio sebagai modul komunikasi. Setup dilakukan dengan memanggil modul-modul yang dibutuhkan ke dalam aplikasi javascript dan mempersiapkan komponen yang dibutuhkan. Rutin setup sistem ditampilkan pada gambar 4 berikut ini.

```
const express = require('express');
const app = express();
const http = require('http');
const { Server } = require("socket.io");
const server = http.createServer(app);
const io = new Server(server, {
  cors: {
    origin: "*",
    methods: ["GET", "POST"],
  }
});
```

Gambar 4. Setup Sistem

Setup sistem di atas menghasilkan objek app sebagai penampil halaman web dan object io yang digunakan untuk komunikasi antar komponen. Konfigurasi ditambahkan ke socketio untuk mengatasi kebijakan *Cross Origin Resource Sharing* (CORS).

Protokol Komunikasi

Protokol komunikasi mendefinisikan format data yang dikirimkan oleh aggregator ke client dan respon yang dikirimkan oleh client ke aggregator. Setiap komponen harus mengikuti format ini agar komunikasi berjalan lancar. Format data yang digunakan adalah *Javascript Object Notation* (JSON) dengan ketentuan terlampit pada gambar 5 berikut.

```
// query (agregator)
{"query": "nama_katalog"}

// Response (client)
{
  "toko": "nama_toko",
  "telepon": "telepon_toko",
  "email": "email_toko",
  "website": "http://umkm1.example.com",
  "nama_item": "nama_item",
  "stok": "jumlah_stok",
  "foto": "foto_base64"
}
```

Gambar 5. Protokol Komunikasi

Aggregator

Aggregator adalah aplikasi web utama yang diakses oleh masyarakat untuk mencari katalog produk UMKM. Aplikasi ini menampilkan kotak pencarian dan antar muka hasil pencarian. Aplikasi web dilayani oleh webserver expressjs yang

menyediakan halaman html statis. Kode web server ditampilkan pada gambar 6 di bawah ini.

```
app.get('/agregator', (req, res) => {
  res.sendFile(__dirname + '/agregator.html');
});
```

Gambar 6. ExpressJS Menyediakan Halaman Agregator

Potongan kode pada gambar 6 di atas bertugas menyediakan halaman statis yaitu agregator.html . Halaman ini berisi kode HTML untuk menampilkan kotak pencarian dan hasil pencarian. Halaman aggregator juga memiliki kode javascript untuk untuk berkomunikasi dengan hub. Kode aggregator diperlihatkan pada gambar 7 berikut ini.

```
<!-- Kode HTML tampilan -->
<span style="text-align: center;"><h3>Agregator Kalatog UMKM</h3></span>
<span style="text-align: center;">
  <form id="form" action="">
    <input id="query" autocomplete="off" /><button>Query</button>
  </form>
</span>
<div id="katalog"></div>

<!-- Kode Javascript -->
<script src="socket.io.min.js"></script>
<script src="minisearch.js"></script>
<script>
  // koneksi ke socket io dengan token
  var socket = io({auth: {token: "t-agregator"}});
  // mengirimkan query
  var form = document.getElementById('form');
  var input = document.getElementById('query');

  form.addEventListener('submit', function(e) {
    e.preventDefault();
    if (input.value) {
      socket.emit('query', `{"query": "${input.value}"}`);
      input.value = '';
    }
    // kosongkan box hasil pencarian:
    let katalogDiv = document.getElementById('katalog');
    katalogDiv.innerHTML = "";
  });
</script>
```

Gambar 7. Kode Agregator

Agregator melakukan koneksi ke hub menggunakan class “io” dengan parameter “auth” berisi token untuk melakukan otorisasi. Kode token ‘t-agregator’ adalah *private shared key* yang berarti disepakati sebelumnya antara hub dan aggregator.

Agregator menanti input pencarian dari user yang selanjutnya dikirimkan ke hub untuk di-broadcast ke semua client katalog yang terhubung. Input tersebut adalah kode {query: input_pencarian} yang sebelumnya didefinisikan pada protocol komunikasi.

Fungsi aggregator selanjutnya adalah menerima respon hasil pencarian dan menampilkannya kepada user. Gambar 8 di bawah menampilkan kode yang bertugas menerima hasil query dan menampilkannya pada halaman web.

```
socket.on('res-query', function(msg) {
  msg.results.forEach((item) => {
    let katalogDiv = document.getElementById('katalog');
    katalogDiv.innerHTML += '<p><strong>'+item.nama_item+'</strong><br>';
    katalogDiv.innerHTML += 'Stok :'+item.stok+'<br>';
    katalogDiv.innerHTML += 'Gambar : <br>';
    katalogDiv.innerHTML += 'Toko :'+item.toko+'<br>';
    katalogDiv.innerHTML += 'Telepon :'+item.telepon+'<br>';
    katalogDiv.innerHTML += 'Email :'+item.email+'<br>';
    katalogDiv.innerHTML += 'Website :'+item.website+'<br>';
    katalogDiv.innerHTML += '</p>';
  });
});
```

Gambar 8. Kode Penampil Hasil Pencarian Oleh Agregator

Potongan kode pada gambar 8 di atas memonitor broadcast dengan pengenal 'res-query'. Jika broadcast diterima, maka akan menampilkan struktur data hasil pencarian katalog pada halaman web aggregator.

HUB

Hub merupakan aplikasi server yang menjembatani komunikasi antara aggregator dan client katalog. Aplikasi ini mengelola dan menerapkan kebijakan kepada komponen aggregator dan client yang terhubung padanya. Hub melakukan beberapa tugas utama yaitu melakukan otorisasi agregator menggunakan token yang sudah ditentukan, menerima permintaan query dari aggregator dan meneruskannya (broadcast) ke semua client katalog, selanjutnya menerima respon query yang berupa hasil pencarian katalog dan meneruskannya hanya ke aggregator.

Potongan kode proses otorisasi ditampilkan pada gambar 9 di bawah. Potongan kode tersebut mencari objek "auth" yang dikirimkan oleh komponen yang melakukan koneksi seperti terlihat pada gambar 7 di atas. Jika mendapatkan koneksi yang menyertakan objek auth, hub akan mencocokkan nilainya dengan nilai yang terdefinisi sebelumnya. Jika cocok, maka koneksi tersebut akan dimasukkan ke room "r_agregator". Room ini adalah room khusus aggregator sehingga hub dapat mengirimkan pesan pribadi yang berisi hasil pencarian katalog ke aggregator.

```
var a_token = 't-agregator';
const token = socket.handshake.auth.token;
if(token == a_token)
{
  socket.join('r_agregator');
}
```

Gambar 9. Proses Otorisasi

Fungsi hub selanjutnya adalah menjembatani komunikasi antara aggregator dan client. Hub dapat menerima broadcast dan meneruskannya ke setiap node yang terhubung kepadanya. Hub bekerja dengan memonitor kedatangan *event*. Komponen aggregator menggunakan event 'query' untuk mengirimkan permintaan query. Hub memonitor event 'query' dan mem-broadcast pesan yang dikirimkan melalui event 'req-query' kepada semua komponen client yang juga memonitor event yang sama. Client memproses permintaan pada event 'req-query' dengan melakukan pencarian katalog dan mengirimkan kembali hasil pencarian tersebut melalui event 'res-query' yang dimonitor oleh hub. Selanjutnya hub membroadcast pesan dari 'res-query' melalui room 'r_agregator' yang hanya berisikan aggregator untuk mengirimkan pesan tersebut secara privat hanya

kepada agregator. Gambar 10 berikut ini menampilkan kode proses jembatan komunikasi antar komponen oleh hub.

```
// agregator query data:
socket.on('query', (msg) => {
  // query ke semua klien:
  console.log('Query: ' + msg)
  io.emit('req-query', msg);
});

// katalog memberikan respons:
socket.on('res-query', (msg) => {
  console.log('Respon: ' + msg)
  // kirim balasan ke agregator saja:
  socket.to('r_agregator').emit('res-query', msg);
});
```

Gambar 10. Proses Komunikasi Yang Dijembatani Hub.

Client Katalog

Client katalog merupakan aplikasi yang berada pada sisi UMKM. Komponen ini bertugas untuk menghimpun katalog produk UMKM dan melayani permintaan pencarian katalog produk oleh aggregator. Client terhubung kepada aggregator melalui perantara hub. Banyak client dapat terhubung ke hub, namun setiap client katalog memiliki alur kerja yang sama.

Agar dapat bekerja dengan baik, sebuah client katalog harus memiliki daftar katalog produk, selanjutnya memiliki mekanisme pencarian produk dan mendapatkan akses untuk terhubung ke hub. Katalog produk dapat berupa basis data maupun struktur data. Penelitian ini menggunakan struktur data object javascript sebagai penyimpanan data katalog dan pustaka minisearch untuk melakukan pencarian data katalog. Gambar 11 berikut menampilkan penggunaan struktur data dan modul pencarian.

```
<script src="socket.io.min.js"></script>
<script src="minisearch.js"></script>
// informasi toko:
var toko = "UMKM 2";
var telepon = "08136354321";
var email = "umkm2@example.com";
var website = "http://umkm2.example.com";
// data katalog:
const katalog = [
  {"id": "1", "nama_item": "Sulam Benang Emas", "stok": "10", "foto": ""},
  {"id": "2", "nama_item": "Tembikar", "stok": "8", "foto": ""},
  {"id": "3", "nama_item": "Randang baluik", "stok": "19", "foto": ""},
];
// minisearch
let miniSearch = new MiniSearch({
  fields: ['nama_item'], // field yang diindeks untuk pencarian
  storeFields: ['nama_item', 'stok', 'foto'] // fields hasil pencarian
});
// Index semua katalog
miniSearch.addAll(katalog);
```

Gambar 11. Struktur Data Katalog UMKM

Setelah data katalog dan modul pencarian didefinisikan, berikutnya komponen client harus terhubung di hub dan memonitor permintaan pencarian katalog. Gambar 12 di bawah menampilkan kode pencarian katalog dan respon permintaan pencarian katalog.

```
var socket = io("http://localhost:3000",{auth: {token : ""}});
var term_query = "";
socket.on('req-query', function(msg) {
  let reqq = JSON.parse(msg);
  term_query = reqq.query;
  console.Log(term_query);
  let results = miniSearch.search(term_query, { fuzzy: 0.1 });
  console.Log(results);
  // rakit objek hasil pencarian:
  let obj_resp = {results: []};
  if (results.length !== 0) {
    results.forEach((item) => {
      obj_resp.results.push(
        {
          "toko": toko,
          "telepon": telepon,
          "email": email,
          "website": website,
          "nama_item": item.nama_item,
          "stok": item.stok,
          "foto": item.foto,
        }
      );
    });
  }
  socket.emit('res-query', obj_resp);
});
```

Gambar 12. Kode Pencarian dan Respon Permintaan Pencarian

Gambar 12 di atas menampilkan potongan kode komponen client. Baris pertama adalah kode koneksi client ke hub. Berikutnya client akan melakukan monitoring event 'req-query' yang berisi permintaan pencarian katalog. Selanjutnya client melakukan pencarian katalog memanfaatkan pustaka minisearch. Hasil pencarian akan diatur kembali agar sesuai dengan spesifikasi yang dibahas pada bab 3.2. Pada proses terakhir, client mengirimkan hasil pencarian melalui event 'res-query' kepada hub, untuk selanjutnya dikirimkan kepada agegator.

C. Hasil dan Pembahasan

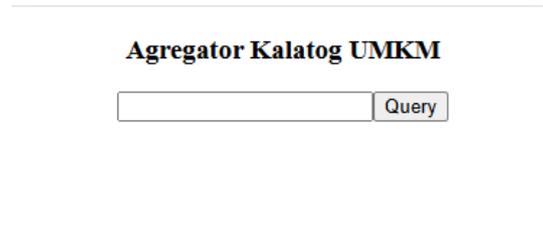
Keluaran dari penelitian ini adalah model dan aplikasi terdistribusi. Terdapat empat komponen yang disiapkan untuk pembahasan yaitu sebuah hub, sebuah aplikasi agegator dan dua buah aplikasi client katalog. Aplikasi hub adalah aplikasi server yang harus dijalankan pertama sekali. Tampilan aplikasi hub diperlihatkan pada gambar 13 berikut ini.

```
C:\hsammir\node\fdmp>node index.js
listening on *:3000
```

Gambar 13. Aplikasi Hub

Apikasi hub bekerja di belakang layar sesuai dengan prosedur yang sudah diprogramkan sebelumnya. Aplikasi ini tidak dapat dikendalikan secara aktif dan hanya menampilkan catatan aktifitas saja.

Aplikasi berikutnya adalah aplikasi agregator yang merupakan aplikasi web yang diakses melalui browser. Halaman aplikasi agregator menampilkan kotak pencarian yang dapat digunakan user untuk mencari katalog produk. Hasil pencarian katalog akan ditampilkan di bawah kotak pencarian. Gamabar 14 dan 15 berikut ini.



Gambar 14. Agregator dan Kotak Pencarian



Gambar 15. Tampilan Hasil Pencarian

Aplikasi terakhir dari keluaran penelitian ini adalah aplikasi client katalog. Aplikasi ini digunakan oleh masing-masing UMKM untuk mempublikasikan katalog produknya sehingga aplikasi ini dapat berjumlah lebih dari satu. Aplikasi client katalog bekerja secara otomatis dengan merespon permintaan pencarian produk pada katalog nya. Penelitian ini mengelola daftar katalog secara langsung pada objek katalog yang berada pada kode aplikasi sehingga tampilan aplikasi hanyalah sebagai penanda bahwa aplikasi client katalog sedang berjalan. Tampilan aplikasi client katalog dieperlihatkan pada gambar 16 berikut.

Kalatog UMKM

Gambar 16. Tampilan Aplikasi Client Katalog

D. Kesimpulan

Penelitian ini berhasil menciptakan metoda komunikasi aplikasi terdistribusi yang mampu mendistribusikan beban aplikasi antara komponennya. Aplikasi yang dihasilkan, khususnya aplikasi client tidak membutuhkan sumber daya besar untuk beroperasi. Model yang dihasilkan dari penelitian ini juga dapat diimplementasikan menggunakan berbagai pustaka software lainnya untuk memberikan fitur yang lebih canggih.

Penelitian selanjutnya bisa meningkatkan fitur pada sisi client dengan memberikan kemudahan dalam mengelola data katalog atau dengan mengintegrasikan data katalog dengan database aplikasi penjualan, jika sudah tersedia sebelumnya

E. Daftar Pustaka

- [1] V. Indiarma, "Inovasi Teknologi Informasi dan Komunikasi di Usaha Mikro Kecil Menengah Pedesaan (Adopsi Inovasi dalam Pemanfaatan Teknologi Informasi dan Komunikasi pada UMKM di Desa Sukasari, Seluma)," *Tuturlogi*, vol. 4, no. 3, p. 90, 2023, doi: 10.21776/ub.tuturlogi.2023.004.03.4.
- [2] S. A. SAPUTRI, I. BERLIANA, I. BERLIANA, and M. F. NASRIDA, "Peran Marketplace Dalam Meningkatkan Daya Saing Umkm Di Indonesia," *Knowl. J. Inov. Has. Penelit. dan Pengemb.*, vol. 3, no. 1, pp. 69–75, 2023, doi: 10.51878/knowledge.v3i1.2199.
- [3] N. S. Prameswari, M. Krisnawati, P. B. Widagdo, and K. H. Luthfia, "Desain E-Katalog Umkm Fesyen 'Womanpreneur Community' Surakarta Dalam Transformasi Ekonomi Digital," *Fash. Fash. Educ. J.*, vol. 12, no. 2, pp. 39–53, 2023, doi: 10.15294/ffej.v12i2.74315.
- [4] N. W. Rustriani, N. P. N. Anggraini, and I. K. S. B. Satwam, "Perancangan Katalog Produk Untuk Meningkatkan Penjualan Umkm," *J. Masy. Mandiri*, vol. 5, no. 5, pp. 2615–2624, 2021, [Online]. Available: <http://journal.ummat.ac.id/index.php/jmm>.
- [5] T. Sudrartono *et al.*, "Kewirausahaan UMKM di era Digital," 2022.
- [6] V. R. Raditya, D. Primasari, and R. Widianingsih, "Analisis Penggunaan Teknologi Aplikasi Akuntansi Pada Usaha Kecil Dan Menengah (Ukm) Di Kabupaten Banyumas Melalui Pendekatan Technology Acceptance Model (Tam)," *J. Ris. Akunt. Soedirman*, vol. 1, no. 1, 2022, doi: 10.32424/1.jras.2022.1.1.6482.
- [7] R. Satria, I. Ahmad, and R. D. Gunawan, "Rancang Bangun E-Marketplace Berbasis Mobile Untuk Meningkatkan Pelayanan Penjualan," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 4, no. 1, pp. 89–95, 2023, doi: 10.33365/jatika.v4i1.2457.
- [8] R. Irawati and I. B. Prasetyo, "Pemanfaatan Platform E-Commerce Melalui Marketplace Sebagai Upaya Peningkatan Penjualan dan Mempertahankan Bisnis di Masa Pandemi (Studi pada UMKM Makanan dan Minuman di Malang)," *J. Penelit. Manaj. Terap.*, vol. 6, no. 2, pp. 114–133, 2021.
- [9] R. N. J. Saputri, B. Nadiya, T. Anjarwati, A. Septianti, and O. Reviandani, "Peran Dinas Koperasi Dalam Optimalisasi e-Katalog Sebagai Sarana Pemberdayaan UMKM di Sidoarjo," *Futur. Acad. J. Multidiscip. Res. Sci. Adv.*, vol. 3, no. 2, pp. 1008–1036, 2025, doi: 10.61579/future.v3i2.519.
- [10] A. Soni, A. Dalal, K. Chaurasia, and M. P. Singh, "Emergency Response App for Enhancing Location-Based Services Using Socket.io and WebRTC," *IEEE Int. Conf. Electr. Electron. Commun. Comput. ELEXCOM 2023*, no. October 2023, 2023, doi: 10.1109/ELEXCOM58812.2023.10370056.
- [11] I. P. A. E. Pratama and I. W. G. Arisna, "Fighter Plane Online Game Based on Design Science Research Methodology Using Socket.io and Nodejs," *Indones. J. Eng. Sci.*, vol. 3, no. 2, pp. 029–037, 2022, doi: 10.51630/ijes.v3i2.38.

- [12] A. Pandiri, S. Shreyas Venishetty, A. Reddy Modugu, and K. V. Sharma, "Scalable and Secure Real-Time Chat Application Development Using MERN Stack and Socket.io for Enhanced Performance 1*," *Front. Collab. Reasearch*, vol. 2, no. 3, pp. 11–22, 2024.