

“Perancangan Sistem Antrian Barbershop Berbasis Web dengan Integrasi AI Chatbot untuk Meningkatkan Kualitas Layanan Pelanggan”

Agusmia Intan Sari – 23343027

Restu Anugrah Prasetyo – 23343016

Program Studi Informatika, Departemen Teknik Elektronika

Fakultas Teknik

Universitas Negeri Padang

Abstrak

Perkembangan teknologi informasi telah mendorong berbagai sektor usaha, termasuk barbershop, untuk mengadopsi sistem digital sebagai upaya peningkatan kualitas layanan. Permasalahan yang sering dijumpai pada barbershop konvensional meliputi pengelolaan antrean yang kurang terstruktur, waktu tunggu pelanggan yang relatif lama, serta keterbatasan akses informasi layanan. Penelitian ini bertujuan untuk merancang sistem antrean barbershop berbasis web yang terintegrasi dengan AI chatbot sebagai media interaksi dan penyampaian informasi kepada pelanggan. Metode pengembangan sistem meliputi tahapan analisis kebutuhan, perancangan sistem, serta implementasi dengan memanfaatkan teknologi web dan basis data. Hasil penelitian menunjukkan bahwa penerapan sistem antrean digital yang dilengkapi AI chatbot mampu meningkatkan efisiensi pengelolaan antrean serta mempermudah pelanggan dalam memperoleh informasi layanan. Dengan demikian, sistem yang dikembangkan diharapkan dapat mendukung proses pelayanan barbershop yang lebih efektif, terorganisasi, dan responsif terhadap kebutuhan pelanggan.

Kata kunci: sistem antrian, barbershop, web, AI chatbot, layanan pelanggan

I. PENDAHULUAN

Perkembangan teknologi informasi saat ini telah memberikan pengaruh yang signifikan terhadap berbagai aspek kehidupan, termasuk pada sektor usaha kecil dan menengah (UKM). Salah satu jenis usaha yang banyak diminati oleh masyarakat adalah barbershop. Namun, dalam pelaksanaannya, masih banyak barbershop yang mengelola antrean secara manual sehingga menimbulkan berbagai permasalahan, seperti antrean yang kurang tertata, waktu tunggu pelanggan yang relatif lama, serta menurunnya tingkat kenyamanan pelanggan.

Pemanfaatan sistem informasi berbasis web menjadi salah satu alternatif solusi untuk mengatasi permasalahan tersebut. Selain itu, penerapan teknologi kecerdasan buatan (Artificial Intelligence) berupa chatbot dapat membantu pelanggan dalam memperoleh informasi secara cepat dan interaktif tanpa harus bergantung langsung pada petugas. Oleh karena itu, diperlukan pengembangan sistem antrean barbershop berbasis web yang terintegrasi dengan AI chatbot guna meningkatkan kualitas pelayanan serta efisiensi operasional barbershop.

II. METODOLOGI

Metodologi penelitian yang diterapkan dalam penulisan artikel ini bertujuan untuk menguraikan tahapan-tahapan dalam perancangan dan pengembangan sistem antrean barbershop berbasis web yang terintegrasi dengan AI chatbot. Metodologi disusun secara sistematis agar proses pengembangan sistem dapat dilaksanakan secara terstruktur serta selaras dengan tujuan penelitian yang telah ditetapkan.

2.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam studi ini adalah penelitian terapan (*applied research*), yaitu penelitian yang menitikberatkan pada pemanfaatan ilmu pengetahuan dan teknologi untuk menyelesaikan permasalahan yang bersifat praktis. Permasalahan yang dikaji dalam penelitian ini adalah pengelolaan antrean barbershop yang masih dilakukan secara manual sehingga belum berjalan secara efisien.

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode *Waterfall*, karena metode ini memiliki tahapan yang jelas dan terstruktur, sehingga cocok digunakan dalam pengembangan sistem informasi berskala kecil hingga menengah.

Tahapan metode *Waterfall* dalam penelitian ini meliputi:

a. Analisis Kebutuhan

Pada tahap ini dilakukan identifikasi kebutuhan sistem melalui analisis terhadap permasalahan yang muncul dalam proses pengelolaan antrean barbershop. Analisis kebutuhan melibatkan dua pihak utama, yaitu pelanggan dan pengelola barbershop. Adapun kebutuhan sistem yang dihasilkan pada tahap ini meliputi:

1. Sistem mampu mengelola data antrean secara digital.
2. Pelanggan dapat melakukan pengambilan nomor antrean secara daring.
3. Sistem dapat menampilkan informasi status antrean secara *real-time*.
4. Tersedianya AI chatbot sebagai sarana interaksi dan penyedia informasi bagi pelanggan.
5. Tersedianya fitur chat dengan admin sebagai sarana komunikasi dengan admin jarak jauh.

b. Perancangan Sistem

Tahap perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah diperoleh. Perancangan mencakup penyusunan arsitektur sistem, perancangan basis data, serta perancangan antarmuka pengguna (*user interface*). Sistem dirancang berbasis web agar dapat diakses dengan mudah melalui berbagai perangkat, seperti komputer maupun smartphone.

c. Implementasi Sistem

Tahap implementasi merupakan proses merealisasikan hasil perancangan ke dalam bentuk sistem yang dapat dioperasikan. Pada tahap ini dilakukan pengembangan perangkat lunak melalui pengkodean menggunakan teknologi web serta integrasi AI chatbot ke dalam sistem antrean. Sistem yang dikembangkan memungkinkan pelanggan untuk berinteraksi dengan chatbot guna memperoleh informasi terkait status antrean dan layanan barbershop.



Gambar 1. Arsitektur dan Alur Kerja Sistem Antrian Barbershop Berbasis Web

Gambar 1 menampilkan arsitektur serta alur kerja dari sistem antrean barbershop berbasis web yang dikembangkan. Alur proses diawali dengan pelanggan yang melakukan pendaftaran antrean melalui antarmuka web. Data pendaftaran tersebut selanjutnya diproses oleh sistem dan disimpan ke dalam basis data sebagai informasi antrean.

Sistem kemudian menampilkan status dan urutan antrean secara otomatis kepada pelanggan maupun pengelola barbershop secara *real-time*. Selain itu, AI chatbot diintegrasikan ke dalam sistem untuk memberikan layanan informasi secara otomatis kepada pelanggan, seperti status antrean, jadwal layanan, serta menjawab pertanyaan umum. Integrasi antara antarmuka web, sistem, basis data, dan AI chatbot memungkinkan proses pengelolaan antrean berjalan lebih terstruktur, efisien, dan interaktif.

1) Koneksi Database (MongoDB)

```
const mongoose = require('mongoose');
const uri = 'mongodb://127.0.0.1:27017/barberflow';
mongoose.connect(uri, { useNewUrlParser: true, useUnifiedTopology: true });

const connection = mongoose.connection;
connection.on('connected', () => {
  console.log(`Berhasil terhubung ke MongoDB!`);
});
connection.on('error', (err) => {
  console.error(`Terjadi kesalahan saat terhubung ke MongoDB: ${err.message}`);
});
```

Gambar 2. Menghubungkan aplikasi ke database MongoDB.

`mongoose.connect(mongoUri, opts)` merupakan fungsi utama untuk menghubungkan aplikasi Node.js dengan database MongoDB, di mana mongoUri berisi alamat database yang biasanya diambil dari environment variable dan opts berisi opsi konfigurasi koneksi tambahan. Fungsi ini mengembalikan *Promise* sehingga dapat ditangani dengan `.then()` dan `.catch()`, di mana koneksi yang berhasil akan menampilkan pesan sukses, sedangkan kegagalan akan menampilkan pesan error. Mekanisme ini memastikan aplikasi hanya berjalan ketika koneksi ke database berhasil, karena tanpa koneksi tersebut aplikasi tidak

dapat melakukan proses penyimpanan, pengambilan, maupun pengelolaan data.

2) Model Database



```
model? > antrian.js
1 const mongoose = require('mongoose');
2
3 const antrianSchema = new mongoose.Schema();
4
5 antrianSchema.add({
6   nomor_antrian: {
7     type: String,
8     required: true,
9     unique: true,
10    trim: true,
11    index: true
12  },
13   nama_layanan: {
14     type: String,
15     required: true,
16     unique: true,
17     lowercase: true,
18     trim: true,
19     index: true
20  },
21   email: {
22     type: String,
23     required: true,
24     unique: true,
25     lowercase: true,
26     trim: true,
27     index: true
28  },
29   tgl_lahir: {
30     type: Date,
31     default: Date.now,
32     index: true
33  },
34   gender: {
35     type: String,
36     enum: ['laki-laki', 'wanita'],
37     default: 'laki-laki'
38  },
39   tanggal_daftar: {
40     type: Date,
41     default: Date.now,
42     index: true
43  },
44   updated_at: {
45     type: Date,
46     default: Date.now
47  }
48 }, { timestamps: true });
49
50 const antrian = models.Antrian || model('Antrian', antrianSchema);
51
52 module.exports = antrian;
```

Gambar 3. Struktur data untuk antrian.

Schema ini menyimpan informasi seperti nomor antrian yang bersifat unik dan terindeks, nama pelanggan, email, layanan yang dipilih, barber, serta status antrian dengan nilai terbatas (menunggu, dilayani, selesai, batal). Selain itu, schema juga mencatat waktu pembuatan dan pembaruan data melalui field created_at dan updated_at. Middleware pre('save') digunakan untuk memperbarui waktu updated_at setiap kali data disimpan. Model Antrian kemudian diekspor agar dapat digunakan kembali dalam proses penyimpanan, pengambilan, dan pengelolaan data antrian pada aplikasi.



```
model? > user.js
1 const mongoose = require('mongoose');
2
3 const userSchema = new Schema();
4
5 userSchema.add({
6   username: {
7     type: String,
8     required: true,
9     unique: true,
10    trim: true
11  },
12   password: {
13     type: String,
14     required: true
15  },
16   roles: {
17     type: String,
18     enum: ['admin', 'user'],
19     default: 'user'
20  },
21   created_at: {
22     type: Date,
23     default: Date.now
24  },
25   updated_at: {
26     type: Date,
27     default: Date.now
28  },
29   timestamps: true
30 });
31
32 const user = models.User || model('User', userSchema);
33
34 module.exports = user;
```

Gambar 4. Struktur data untuk user/admin.

Schema ini menyimpan informasi username yang bersifat unik dan wajib diisi, password sebagai data autentikasi, serta role pengguna yang dibedakan antara *admin* dan *user* dengan nilai default *user*. Selain itu, terdapat pencatatan waktu pembuatan data

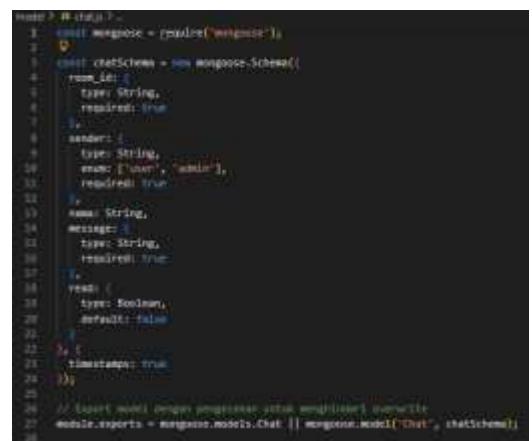
melalui created_at dan fitur timestamps untuk mengelola waktu secara otomatis. Model User kemudian diekspor agar dapat digunakan dalam proses autentikasi dan manajemen pengguna pada aplikasi.



```
model? > pelanggan.js
1 const mongoose = require('mongoose');
2
3 const pelangganSchema = new mongoose.Schema();
4
5 pelangganSchema.add({
6   nome: {
7     type: String,
8     required: true,
9     trim: true
10  },
11   email: {
12     type: String,
13     required: true,
14     unique: true,
15     lowercase: true,
16     trim: true,
17     index: true
18  },
19   nome_sapuan: {
20     type: String,
21     optional: true
22  },
23   alamat: {
24     type: String,
25     optional: true
26  },
27   telp: {
28     type: String,
29     enum: ['081234567890', '081234567891'],
30     default: '081234567890'
31  },
32   tanggal_daftar: {
33     type: Date,
34     default: Date.now,
35     index: true
36  },
37   updated_at: {
38     type: Date,
39     default: Date.now
40  }
41 }, { timestamps: true });
42
43 const pelanggan = models.Pelanggan || model('Pelanggan', pelangganSchema);
44
45 module.exports = pelanggan;
```

Gambar 5. Struktur data untuk pelanggan.

Schema ini memuat informasi utama seperti nama dan email yang wajib diisi, dengan email bersifat unik dan terindeks untuk mempercepat pencarian. Selain itu, disediakan data tambahan berupa nomor telepon dan alamat yang bersifat opsional, status pelanggan (aktif atau inaktif), serta pencatatan waktu pendaftaran dan pembaruan data. Index teks pada field nama dan email digunakan untuk mendukung fitur pencarian yang lebih efisien. Model Pelanggan kemudian diekspor agar dapat digunakan dalam pengelolaan data pelanggan pada aplikasi.



```
model? > chat.js
1 const mongoose = require('mongoose');
2
3 const chatSchema = new mongoose.Schema();
4
5 chatSchema.add({
6   room_id: {
7     type: String,
8     required: true
9  },
10   sender: {
11     type: String,
12     enum: ['User', 'Admin'],
13     required: true
14  },
15   room: {
16     type: String,
17     required: true
18  },
19   message: {
20     type: String,
21     required: true
22  },
23   read: {
24     type: Boolean,
25     default: false
26  },
27   timestamps: true
28 });
29
30 const chat = models.Chat || model('Chat', chatSchema);
31
32 module.exports = mongoose.models.Chat || mongoose.model('Chat', chatSchema);
```

Gambar 6. Struktur data untuk chat.

Schema ini menyimpan informasi ruang chat (`room_id`), pengirim pesan yang dibedakan antara `user` dan `admin`, nama pengirim, isi pesan, serta status pesan apakah sudah dibaca atau belum. Fitur timestamps digunakan untuk mencatat waktu pembuatan dan pembaruan pesan secara otomatis. Model Chat kemudian diekspor dengan pengecekan terlebih dahulu agar tidak terjadi duplikasi model saat digunakan kembali dalam aplikasi.

3) Routing & Endpoint Backend

```
routes> # antrian.js ...
10  /**
11  * Endpoint untuk menambah antrian baru
12  * + POST /api/antrian/add atau /api/antrian/tambah
13  */
14  router.post('/add', 'tambah', async (req, res) => {
15    try {
16      const { name, namePelanggan, email, banner, layanan } = req.body;
17      const nameFinal = name || namePelanggan;
```

Gambar 7. Endpoint untuk tambah antrian.

Endpoint ini digunakan untuk menambahkan data antrian baru ke dalam sistem. Aplikasi akan memvalidasi bahwa nama pelanggan wajib diisi, lalu secara otomatis menentukan nomor antrian berdasarkan nomor terakhir yang tersimpan di database. Data antrian yang berhasil dibuat akan disimpan dengan status awal menunggu, kemudian sistem mengirimkan respons sukses berisi data antrian yang baru ditambahkan.

```
routes> # antrian.js ...
52  /**
53  * Endpoint untuk mengambil daftar antrian
54  * + GET /api/antrian
55  */
56  router.get('/', async (req, res) => {
57    try {
58      // Menggunakan filter status, skip, dan limit
59      const { status, skip = 0, limit = 100 } = req.query;
60      const #filter = status ? { status } : {};
```

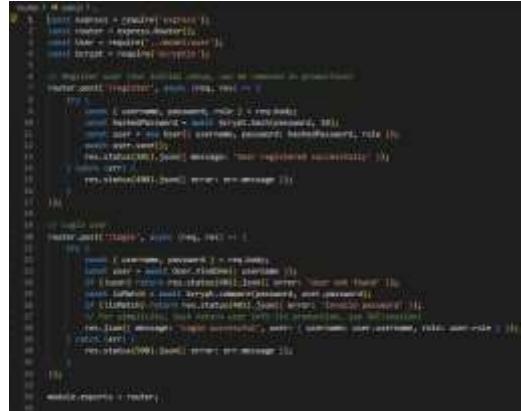
Gambar 8. Endpoint untuk ambil antrian.

Endpoint ini berfungsi untuk mengambil daftar data antrian dari database. Endpoint ini mendukung fitur filter berdasarkan status antrian, serta pagination menggunakan parameter `skip` dan `limit`. Data antrian akan diurutkan berdasarkan waktu pembuatan terbaru, dan respons yang dikirimkan mencakup daftar antrian beserta informasi total data untuk keperluan navigasi halaman.

```
163  /**
164   * DELETE /api/antrian/:id
165   */
166  router.delete('/:id', async (req, res) => {
167    try {
168      const { id } = req.params;
```

Gambar 9. Endpoint untuk hapus antrian.

Endpoint ini digunakan untuk menghapus data antrian berdasarkan ID tertentu. Sebelum penghapusan, sistem memvalidasi format ID untuk mencegah kesalahan query. Jika data antrian berhasil ditemukan dan dihapus, sistem akan mengembalikan respons sukses, sedangkan jika data tidak ditemukan akan mengembalikan pesan error.



```
routes> # user.js ...
1  /**
2  * Endpoint untuk register
3  * + POST /api/register
4  */
5  router.post('/register', async (req, res) => {
6    try {
7      const { username, password } = req.body;
8      const user = await User.create({username, password});
9      res.status(201).json({ message: 'User registered successfully' });
10    } catch (err) {
11      res.status(400).json({ message: err.message });
12    }
13  });

routes> # user.js ...
14  /**
15  * Endpoint untuk login
16  * + POST /api/login
17  */
18  router.post('/login', async (req, res) => {
19    try {
20      const { username, password } = req.body;
21      const user = await User.findOne({username, password});
22      if (!user) {
23        return res.status(401).json({ message: 'Incorrect credentials' });
24      }
25      const token = jwt.sign({ user }, process.env.JWT_SECRET);
26      res.status(200).json({ token });
27    } catch (err) {
28      res.status(400).json({ message: err.message });
29    }
30  });

model> # user.js ...
31  module.exports = User;
```

Gambar 10. Endpoint register & login admin/user.

Endpoint registrasi digunakan untuk menambahkan pengguna baru dengan cara mengenkripsi password menggunakan bcrypt sebelum disimpan ke database agar lebih aman. Sementara itu, endpoint login berfungsi untuk memverifikasi username dan password dengan mencocokkan data yang dimasukkan dengan data terenkripsi di database. Jika autentikasi berhasil, sistem akan mengembalikan informasi dasar pengguna seperti username dan role sebagai tanda bahwa proses login berhasil.

```

routes > js chatbots.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const Antrian = require('../model/antrian');
4
5 /**
6  * POST /api/chatbot
7  * Body: { message: "..." }
8 */
9 router.post('/', async (req, res) => {
10   try {
11     const { message } = req.body;
12     if (!message) {
13       return res.json({ reply: 'Silakan ketik pesan 😊' });
14     }
15
16     const text = message.toLowerCase();
17
18     // =====
19     // HELP
20     // =====
21     if (text.includes('help')) {
22       return res.json({ reply: `
23         🚗 *BarberFlow Bot*
24         Perintah yang bisa kamu gunakan:
25         - antrian sekarang
26         - nomor terakhir
27         - tambah antrian nama [nama]
28         - cek antrian saya [nama]
29         - panggil antrian berikutnya
30         - batalkan antrian [nama]
31
32       `});
33     }
34   }
35 });

```

Gambar 11. Endpoint chat pelanggan dan chatbot.

Endpoint chatbot menggunakan Express yang berfungsi untuk menerima dan memproses pesan dari pengguna melalui metode POST. Endpoint ini membaca pesan dari request body, melakukan validasi agar pesan tidak kosong, lalu mengubah teks menjadi huruf kecil untuk memudahkan pemrosesan perintah. Jika pengguna mengetik perintah help, chatbot akan mengembalikan respons berupa daftar perintah yang tersedia, seperti mengecek antrian, menambah antrian, atau membatalkan antrian. Endpoint ini menjadi dasar interaksi chatbot dalam membantu pengguna mengakses fitur sistem antrian secara otomatis.

4) Socket.IO (Real-time)

```

socket > Mô hình Antrian > node main.js > (socket) > $!<socket>.connection
1
2 // Import model Antrian untuk menyimpan database antrian
3 const Antrian = require('../model/antrian');
4 // Import model Chat untuk menyimpan chat pelanggan dan admin
5 const Chat = require('../model/chat');
6 // Import fungsi Kirim Email ke pelanggan
7 const { kirimEmailPelanggan } = require('../utils/pengiriman');
8
9
10 // Fungsi yang untuk mengambil semua event socket.io
11 module.exports = function (io) {
12   io.on('connection', (socket) => {
13     console.log(`Client Terhubung: ${socket.id}`);
14   });
}

```

Gambar 12. Komunikasi real-time antrian & chat antara client dan server.

Socket.IO berfungsi untuk menangani komunikasi real-time antara client dan server. Model Chat diimporkan untuk menyimpan pesan percakapan antara pelanggan dan admin ke dalam database,

sedangkan fungsi kirimEmailPelanggan digunakan untuk mengirim notifikasi email kepada pelanggan. Fungsi utama ini menerima objek io sebagai parameter dan akan dijalankan ketika server aktif, di mana event connection digunakan untuk mendeteksi saat client berhasil terhubung ke server serta mencatat identitas socket yang terhubung.

d. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fungsi yang dikembangkan berjalan sesuai dengan kebutuhan yang telah ditetapkan. Tahap pengujian mencakup pengujian fungsional sistem, antara lain proses pengambilan nomor antrian, pengelolaan data antrian, serta kemampuan AI chatbot dalam merespons pertanyaan dari pengguna.

2.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini meliputi:

1. Observasi, yaitu pengamatan langsung terhadap proses antrian yang berlangsung di barbershop.
2. Studi Literatur, yaitu pengumpulan data dan referensi dari buku, jurnal, serta sumber ilmiah lainnya yang berkaitan dengan sistem informasi, antrian, dan teknologi AI chatbot.

2.4 Alat dan Teknologi yang Digunakan

Dalam pengembangan sistem antrian barbershop ini digunakan beberapa alat dan teknologi pendukung, antara lain:

1. Node.js, *Runtime environment* untuk menjalankan JavaScript di sisi server (*backend*).
2. Express.js, *Framework* web untuk Node.js yang memudahkan pembuatan REST API dan routing.
3. MongoDB, Database NoSQL untuk penyimpanan data antrian, user, pelanggan, dan chat.
4. Mongoose, ODM (Object Data Modeling) untuk MongoDB, memudahkan interaksi database di Node.js.
5. Socket.IO, Library untuk komunikasi real-time antara server dan client (notifikasi antrian, chat).
6. Bcryptjs, Library untuk enkripsi password user/admin.
7. HTML, CSS, JavaScript, Digunakan untuk membangun tampilan antarmuka web (frontend).

2.5 Alur Penelitian

Alur penelitian dimulai dari identifikasi masalah, dilanjutkan dengan analisis kebutuhan, perancangan sistem, implementasi, serta pengujian sistem. Tahapan-tahapan tersebut dilakukan secara berurutan untuk menghasilkan sistem antrian barbershop berbasis web yang sesuai dengan kebutuhan pengguna dan mampu meningkatkan kualitas layanan pelanggan.

III. HASIL DAN PEMBAHASAN

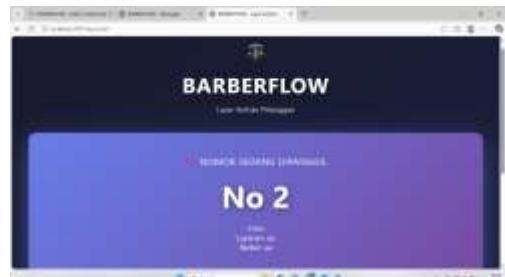
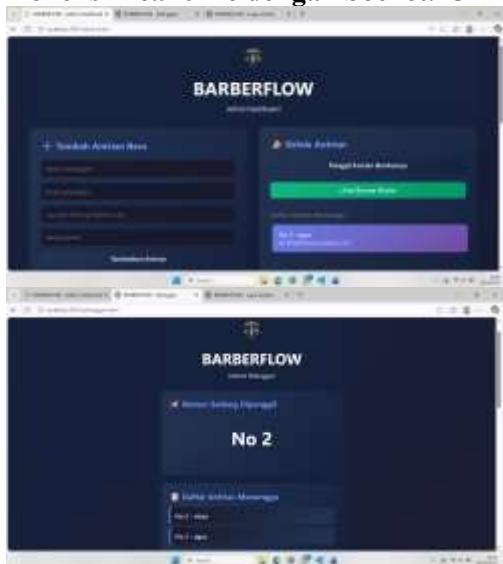
Bagian ini membahas hasil dari bagian perancangan dan implementasi sistem antrian barbershop berbasis web yang terintegrasi dengan AI chatbot, serta analisis terhadap kinerja dan manfaat sistem yang telah dikembangkan.

3.1 Hasil Implementasi Sistem

Hasil penelitian ini berupa pengembangan sistem antrean barbershop berbasis web yang mampu mengelola proses antrean secara digital. Sistem tersebut memungkinkan pelanggan untuk melakukan pengambilan nomor antrean tanpa harus datang langsung ke lokasi barbershop, sehingga dapat meminimalkan penumpukan antrean serta mengurangi waktu tunggu pelanggan.

Sistem yang dikembangkan dilengkapi dengan fitur utama meliputi pendaftaran antrean, pemantauan status antrean, serta pengelolaan antrean oleh pihak barbershop. Data antrean disimpan dalam basis data dan disajikan secara terstruktur, sehingga memudahkan pengelola dalam memantau urutan dan proses pelayanan.

1) Koneksi Real-time dengan Socket.IO



Gambar 13. Sistem berhasil menghubungkan client (pelanggan/admin/layar) ke server secara real-time menggunakan Socket.IO.

Koneksi real-time dengan Socket.IO pada sistem antrian barbershop ini memungkinkan komunikasi dua arah secara langsung antara server dan seluruh client (pelanggan, admin, maupun layar display) tanpa perlu melakukan refresh halaman. Dengan memanfaatkan library Socket.IO, setiap client yang terhubung ke server akan mendapatkan sebuah koneksi socket yang unik. Melalui koneksi ini, server dapat mengirimkan pembaruan data antrian, notifikasi, maupun pesan chat secara instan ke semua client yang membutuhkan. Sebaliknya, client juga dapat mengirimkan permintaan atau pesan ke server, seperti menambah antrian, memanggil antrian berikutnya, atau mengirim chat ke admin. Dengan adanya koneksi real-time ini, seluruh proses antrian dan komunikasi dalam sistem menjadi lebih responsif, efisien, dan user-friendly, karena setiap perubahan status antrian atau pesan akan langsung diterima oleh semua pihak terkait secara otomatis.

2) Manajemen Room Socket



Gambar 14. Admin dapat bergabung ke room admin untuk menerima semua pesan dari pelanggan.

Manajemen room pada Socket.IO di sistem ini memungkinkan setiap client untuk bergabung ke dalam “ruangan” tertentu sesuai peran dan kebutuhannya. Pelanggan yang ingin melakukan komunikasi privat dengan admin akan bergabung ke room khusus berdasarkan identitas atau nomor antrian mereka (misal: pelanggan-7), sehingga pesan yang dikirim hanya akan

diterima oleh admin dan pelanggan tersebut. Sementara itu, admin dapat bergabung ke room “admin-room” yang berfungsi sebagai pusat komunikasi untuk menerima semua pesan dari pelanggan. Dengan konsep room ini, sistem dapat mengelola komunikasi secara terpisah dan efisien, memastikan pesan dan notifikasi hanya diterima oleh pihak yang berkepentingan tanpa mengganggu client lain. Fitur ini sangat penting untuk menjaga privasi, keamanan, dan kenyamanan komunikasi antara pelanggan dan admin dalam sistem antrian barbershop.

3) Notifikasi Email Otomatis



Gambar 15. Ketika antrian dipanggil, sistem secara otomatis mengirimkan email notifikasi ke pelanggan

Fitur notifikasi email otomatis pada sistem ini berfungsi untuk memberikan pemberitahuan secara langsung kepada pelanggan ketika nomor antrian mereka dipanggil. Setiap kali admin memanggil antrian berikutnya dan data antrian tersebut memiliki alamat email yang valid, sistem akan secara otomatis menjalankan fungsi pengiriman email ke pelanggan terkait. Email ini berisi informasi penting seperti nama pelanggan dan nomor antrian yang sedang dipanggil, sehingga pelanggan dapat segera mengetahui giliran mereka tanpa harus terus-menerus memantau layar antrian di barbershop. Dengan adanya notifikasi email otomatis ini, pengalaman pelanggan menjadi lebih nyaman dan efisien, serta meminimalisir risiko pelanggan terlewat atau terlambat mengetahui giliran antrian mereka.

4) Chatbot Otomatis



Gambar 16. Fitur chatbot sederhana yang dapat menjawab pertanyaan umum pelanggan secara otomatis

Fitur chatbot otomatis pada sistem antrian barbershop ini dirancang untuk membantu pelanggan mendapatkan jawaban cepat atas pertanyaan-pertanyaan umum tanpa perlu menunggu respon dari admin. Chatbot akan secara otomatis membalas pesan pelanggan berdasarkan kata kunci tertentu, seperti menanyakan jumlah antrian yang menunggu, nomor antrian yang sedang dipanggil, estimasi waktu tunggu, jam operasional, hingga lokasi barbershop. Dengan logika sederhana yang memanfaatkan pencocokan kata kunci pada pesan yang dikirim pelanggan, chatbot dapat memberikan informasi yang relevan dan responsif. Fitur ini tidak hanya meningkatkan efisiensi pelayanan, tetapi juga memberikan pengalaman interaktif yang lebih baik bagi pelanggan, terutama di luar jam operasional atau saat admin sedang tidak aktif.

3.2 Integrasi AI Chatbot dalam Sistem

AI chatbot yang terintegrasi dalam sistem berperan sebagai asisten virtual bagi pelanggan. Chatbot tersebut mampu menyajikan informasi terkait status antrian, jadwal layanan, serta merespons pertanyaan umum yang sering diajukan oleh pelanggan. Kehadiran chatbot memungkinkan pelanggan memperoleh informasi secara cepat dan interaktif tanpa harus menunggu respons langsung dari petugas barbershop. Integrasi AI chatbot juga membantu mengurangi beban kerja pengelola barbershop, khususnya dalam memberikan informasi yang bersifat rutin. Hal ini berdampak pada peningkatan efisiensi pelayanan serta pengalaman pengguna yang lebih baik.

3.3 Analisis Kinerja Sistem

Berdasarkan hasil pengujian fungsional, sistem antrian yang dikembangkan menunjukkan kinerja yang baik dan telah memenuhi kebutuhan yang telah ditetapkan. Proses pengambilan nomor antrian, penyimpanan data, serta penyajian informasi antrian dapat berjalan tanpa kendala yang signifikan.

Selain itu, respons AI chatbot dalam menyampaikan informasi kepada pelanggan dinilai cepat dan relevan. Sistem juga mampu menampilkan informasi antrian secara *real-time*, sehingga pelanggan dapat memantau perkembangan antrian dengan lebih mudah dan efisien.

3.4 Pembahasan

Penerapan sistem antrean barbershop berbasis web memberikan kontribusi positif terhadap peningkatan proses pelayanan. Jika dibandingkan dengan sistem manual, sistem digital ini mampu menciptakan pengelolaan antrean yang lebih tertata serta meminimalkan potensi kesalahan dalam pencatatan data.

Pemanfaatan AI chatbot sebagai media interaksi juga menjadi keunggulan tambahan dalam sistem yang dikembangkan. Chatbot tidak hanya berperan sebagai penyedia informasi, tetapi juga mampu meningkatkan kepuasan pelanggan melalui layanan yang responsif dan mudah diakses. Dengan demikian, integrasi teknologi web dan AI chatbot terbukti dapat meningkatkan kualitas pelayanan sekaligus efisiensi operasional barbershop.

IV. KESIMPULAN

Berdasarkan hasil perancangan dan pembahasan terhadap sistem antrean barbershop berbasis web yang terintegrasi dengan AI chatbot, dapat disimpulkan bahwa pemanfaatan teknologi informasi mampu memberikan solusi terhadap permasalahan pengelolaan antrean yang sebelumnya dilakukan secara manual. Sistem yang dikembangkan mampu mengelola data antrean secara lebih terstruktur dan efisien, serta memberikan kemudahan bagi pelanggan dalam mengakses informasi layanan.

Integrasi AI chatbot dalam sistem antrean berfungsi sebagai asisten virtual yang mampu menyampaikan informasi kepada pelanggan secara cepat dan interaktif. Kehadiran chatbot ini tidak hanya meningkatkan kenyamanan pelanggan, tetapi juga membantu mengurangi beban kerja pengelola barbershop dalam menangani pertanyaan yang bersifat rutin.

Dengan demikian, penerapan sistem antrean barbershop berbasis web yang terintegrasi dengan AI chatbot terbukti dapat meningkatkan kualitas pelayanan serta efisiensi operasional. Sistem ini diharapkan dapat menjadi alternatif solusi bagi pelaku usaha barbershop dalam memanfaatkan teknologi digital guna meningkatkan daya saing dan kepuasan pelanggan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Randi Proska Sandra, S.Pd., M.Sc., selaku dosen mata kuliah Pemrograman Jaringan, atas bimbingan, arahan, serta ilmu yang telah diberikan selama proses perkuliahan. Berkat penjelasan dan dukungan beliau, penulis dapat memahami materi dengan baik dan menyelesaikan tugas serta proyek perkuliahan dengan maksimal.

REFERENSI

- [1] Laudon, K. C., & Laudon, J. P. (2020). Management Information Systems: Managing the Digital Firm. Pearson Education.
- [2] Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. Artificial Intelligence Applications and Innovations, 373–383.
- [3] Prasetyo, E., & Wibowo, S. (2021). Implementasi sistem antrian online untuk meningkatkan kualitas pelayanan. Jurnal Sistem Informasi, 17(1), 45–54.
- [4] Setiawan, D., & Putri, R. E. (2020). Pemanfaatan teknologi informasi dalam peningkatan layanan usaha kecil menengah. Jurnal Informatika, 14(3), 201–210.
- [5] Følstad, A., Araujo, T., Law, E., Brandtzaeg, P. B., Papadopoulos, S., Reis, L., & Wald, R. (2021). Future directions for chatbot research. Computers in Human Behavior, 123, 106863.
- [6] Nurhayati, N., & Rahman, A. (2021). Pengembangan sistem antrian berbasis web untuk meningkatkan kualitas pelayanan. Jurnal Teknologi Informasi dan Ilmu Komputer, 8(3), 451–458.
- [7] McTear, M., Callejas, Z., & Griol, D. (2020). The Conversational Interface: Talking to Smart Devices. Springer.
- [8] Brandtzaeg, P. B., & Følstad, A. (2020). Why people use chatbots. Internet Interactions, 2(1), 1–7.
- [9] Pratama, A., & Hidayat, R. (2022). Sistem informasi antrian berbasis web dengan notifikasi real-time. Jurnal Informatika dan Komputer, 7(1), 15–24.
- [10] Alshammary, T., & Alsabaei, A. (2023). AI-powered chatbots for customer service: A systematic review. Journal of King Saud University – Computer and Information Sciences, 35(8), 101679.

PERNYATAAN

Dengan ini kami menyatakan bahwa makalah yang kami tulis ini adalah tulisan kami sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi. Apabila nanti ditemukan plagiasi, maka kami bersedia menerima konsekuensi dengan pembatalan nilai sepenuhnya

Padang, 23 Desember 2025

Penulis,

<p>Anggota 1</p>  <p>Agusmia Intan Sari NIM : 23343027</p>	<p>Anggota 2</p>  <p>Restu Anugrah Prasetyo NIM : 23343016</p>
---	--