

LAPORAN PROYEK:

SISTEM KONTAK PENGGUNA

I. PENDAHULUAN

1.1 LATAR BELAKANG

Di era digital saat ini, interaksi antara penyedia layanan (website) dengan pengguna merupakan hal yang sangat krusial. Sebuah website yang baik tidak hanya menyajikan informasi, tetapi juga menyediakan sarana bagi pengguna untuk memberikan umpan balik, pertanyaan, atau pesan lainnya. Formulir kontak adalah jembatan utama untuk komunikasi ini. Namun, seringkali formulir kontak yang ada bersifat statis, kurang interaktif, dan tidak memberikan validasi input secara *real-time*, sehingga dapat mengurangi pengalaman pengguna (UX) dan menyebabkan data yang tidak valid masuk ke sistem.

Proyek "Sistem Kontak Pengguna Interaktif" ini dibangun untuk mengatasi permasalahan tersebut. Dengan mengimplementasikan teknologi sisi klien seperti jQuery untuk validasi dan animasi dinamis, serta teknologi sisi server PHP untuk pemrosesan data, aplikasi ini bertujuan untuk menciptakan sebuah formulir kontak yang tidak hanya fungsional tetapi juga responsif dan menarik secara visual.

1.2 TUJUAN PROYEK

Tujuan utama dari proyek ini adalah:

1. Mengembangkan aplikasi web formulir kontak yang fungsional untuk menerima input berupa nama, email, dan komentar dari pengguna.
2. Mengimplementasikan validasi data di sisi klien (client-side) menggunakan JavaScript dan jQuery untuk memastikan data yang dikirim ke server valid dan sesuai format.
3. Memproses dan menyimpan data yang diterima dari formulir menggunakan PHP Array di sisi server (server-side).
4. Menyediakan pengalaman pengguna yang responsif dengan interaksi dinamis menggunakan jQuery untuk validasi formulir.
5. Membangun antarmuka yang bersih, menarik, dan responsif menggunakan HTML dan CSS

1.3 RUANG LINGKUP PROYEK

Proyek ini memiliki ruang lingkup sebagai berikut:

- Termasuk (In-Scope):
 - Pembuatan halaman formulir dengan input nama, email, dan komentar.
 - Validasi client-side untuk semua input.
 - Pemrosesan data formulir menggunakan PHP.
 - Penyimpanan data sementara menggunakan PHP Array (berbasis sesi).
 - Tampilan halaman konfirmasi yang menunjukkan data yang berhasil dikirim.
 - Implementasi jQuery untuk validasi input formulir di sisi klien.
 - Desain antarmuka yang responsif.
- Tidak Termasuk (Out-of-Scope):
 - Penyimpanan data ke dalam database permanen (seperti MySQL).
 - Fitur pengiriman email otomatis ke administrator.
 - Sistem autentikasi atau login pengguna.
 - Implementasi AJAX untuk pengiriman data (di luar fitur wajib).

II. ANALISIS KEBUTUHAN

2.1 KEBUTUHAN SISTEM

Kebutuhan sistem diidentifikasi menjadi dua kategori, yaitu fungsional dan non-fungsional.

- Kebutuhan Fungsional:
 1. Sistem harus dapat menampilkan formulir dengan field untuk nama, email, dan komentar.
 2. Sistem harus dapat memvalidasi input:
 - Nama dan email tidak boleh kosong.
 - Email harus memiliki format yang valid.

- Komentar harus lebih dari 10 karakter.
- 3. Sistem harus menampilkan pesan error jika validasi gagal.
- 4. Sistem harus dapat memproses data yang dikirim menggunakan metode POST dengan PHP.
- 5. Sistem harus menyimpan setiap data yang masuk ke dalam sebuah PHP Array.
- 6. Sistem harus menampilkan data yang baru saja dikirim pada halaman konfirmasi.
- Kebutuhan Non-Fungsional:
 1. Responsivitas: Tampilan aplikasi harus dapat beradaptasi dengan baik di perangkat desktop maupun mobile.
 2. User Experience: Aplikasi harus memberikan umpan balik visual yang jelas (pesan error, animasi) untuk meningkatkan pengalaman pengguna.
 3. Keamanan: Melakukan sanitasi input dasar di sisi server untuk mencegah serangan sederhana seperti XSS (Cross-Site Scripting).

2.2 KEBUTUHAN PERANGKAT KERAS DAN PERANGKAT LUNAK

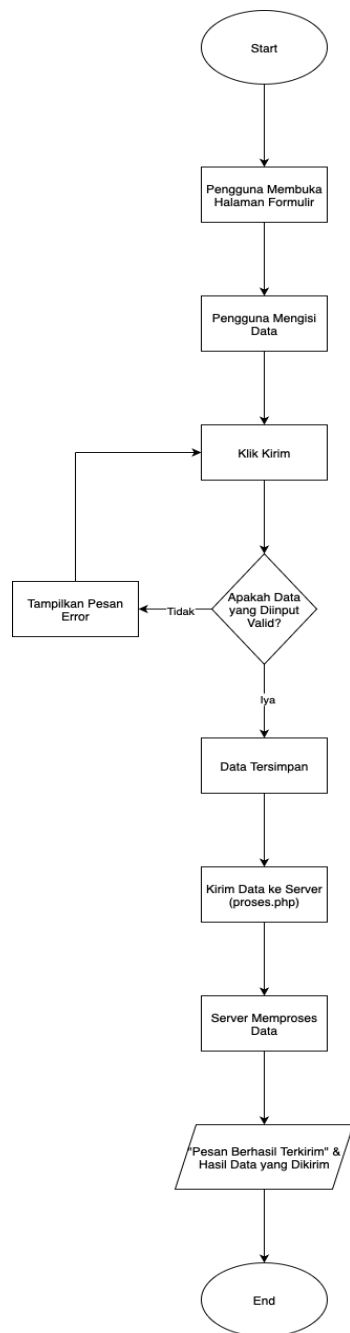
Deskripsi perangkat keras dan perangkat lunak yang digunakan untuk membangun dan menjalankan proyek ini adalah sebagai berikut:

- Perangkat Keras:
 - Laptop: Lenovo IdeaPad S340-14API
 - RAM: 12 GB
- Perangkat Lunak:
 - Sistem Operasi: Windows 11
 - Web Server Environment: Laragon Full 6.0
 - PHP Interpreter: PHP 8.1.10 (bagian dari paket Laragon)
 - Library: jQuery v3.7.1
 - Editor Kode: Visual Studio Code
 - Browser: Google Chrome

III. DESAIN SISTEM

3.1 ARSITEKTUR SISTEM

Berikut adalah flowchart dari Sistem Kontak Pengguna :



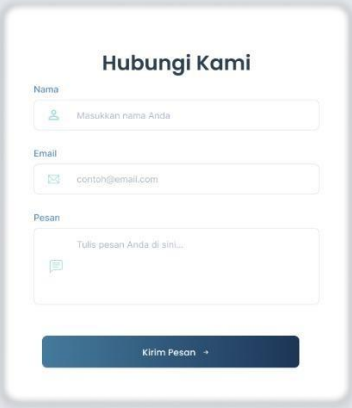
3.2 DESAIN ANTARMUKA PENGGUNA (UI/UX)

Desain figma yang saya buat seperti berikut :

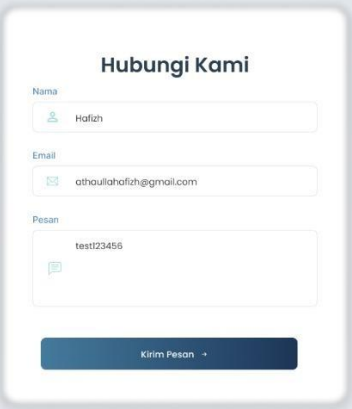
Dengan desain prototype seperti ini :

https://www.figma.com/proto/gxbdUgEfYjKga86yCmwQf2/Sistem_Kontak_Pengguna?page-id=0%3A1&node-id=19-3&p=f&m=draw&scaling=scale-down&content-scaling=fixed&starting-point-node-id=19%3A3&t=nnA8mWitwlbVW9b1-1

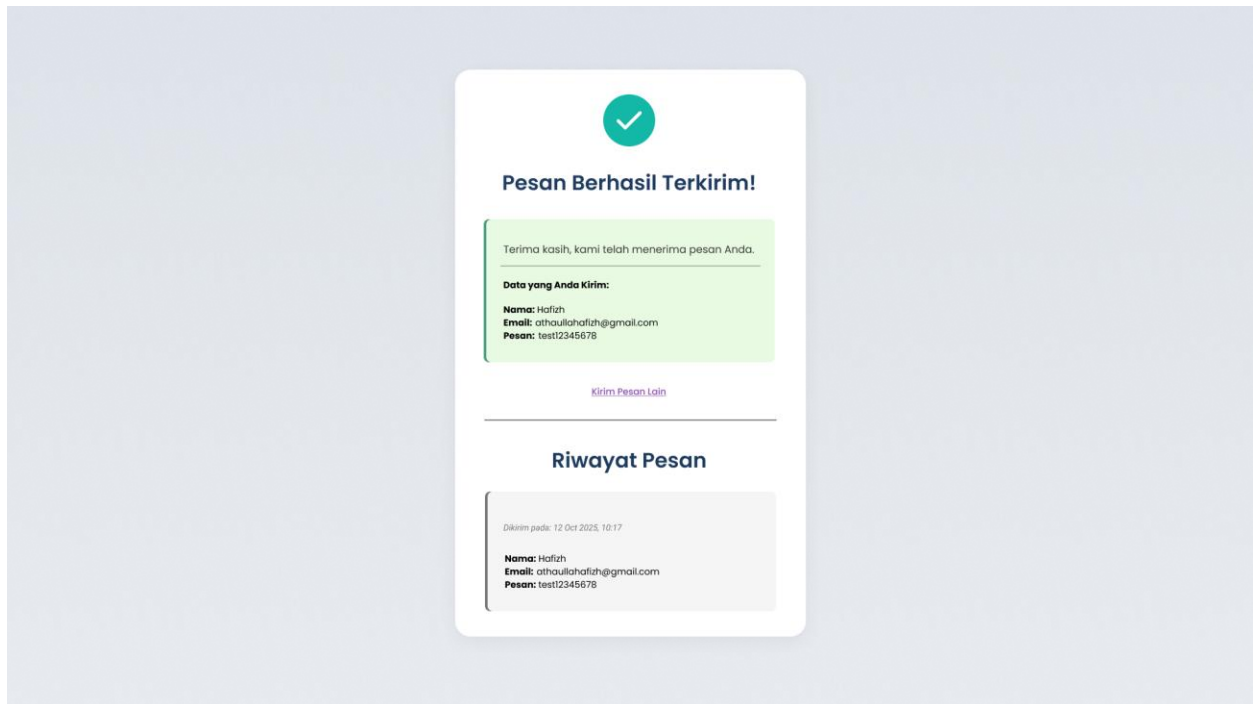
Home Page



Fill Page



Success Page



IV. IMPLEMENTASI SISTEM

4.1 TEKNOLOGI YANG DIGUNAKAN

- Bahasa Pemrograman: HTML, CSS, JavaScript, PHP.
- Library: jQuery v3.7.1.
- Alat Pengembangan: Visual Studio Code

4.2 PROSES PENGEMBANGAN

Berikut adalah penjabaran dari setiap file kode yang digunakan dalam proyek ini.

4.2.1 index.html (Struktur Formulir)

File ini bertanggung jawab untuk membangun struktur dasar dari halaman formulir kontak. Formulir ini berisi tiga input utama: nama, email, dan pesan, serta sebuah tombol kirim. Setiap grup input dibungkus dalam `div.form-group` dan `input field` ditemani oleh ikon SVG untuk estetika. File ini juga memuat `style.css` untuk desain dan file `jQuery` serta `script.js` di bagian bawah untuk fungsionalitas.

```
<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Hubungi Kami</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container">

        <form id="contactForm" action="proses.php" method="POST">

            <h1>Hubungi Kami</h1>

            <div class="form-group">

                <label for="nama">Nama</label>

                <div class="input-with-icon">

                    <svg xmlns="http://www.w3.org/2000/svg"
width="16" height="16" fill="currentColor" class="bi bi-person"
viewBox="0 0 16 16">...</svg>

                    <input type="text" id="nama" name="nama"
placeholder="Masukkan nama Anda">

                </div>

                <div id="namaError" class="error-message"></div>

            </div>

            <div class="form-group">

                <label for="email">Email</label>

                <div class="input-with-icon">

                    <svg xmlns="http://www.w3.org/2000/svg"
width="16" height="16" fill="currentColor" class="bi bi-envelope"
viewBox="0 0 16 16">...</svg>
```

```

        <input type="email" id="email" name="email"
placeholder="contoh@email.com">

    </div>

    <div id="emailError" class="error-message"></div>
</div>

<div class="form-group">

    <label for="pesan">Pesan</label>

    <div class="input-with-icon">

        <svg xmlns="http://www.w3.org/2000/svg"
width="16" height="16" fill="currentColor" class="bi bi-chat-left-
text" viewBox="0 0 16 16">...</svg>

        <textarea id="pesan" name="pesan" rows="5"
placeholder="Tulis pesan Anda di sini..."></textarea>

    </div>

    <div id="pesanError" class="error-message"></div>
</div>

<button type="submit">

    <span>Kirim Pesan</span>

</button>

</form>
</div>

<script src="jquery-3.7.1.js"></script>

<script src="script.js"></script>
</body>
</html>

```


4.2.2 style.css (Desain Visual)

File style.css mengatur seluruh aspek visual dari aplikasi. Desain menggunakan latar belakang warna solid dan fokus utamanya adalah pada tata letak yang rapi menggunakan flexbox, tipografi yang jelas dari font 'Poppins', dan pengalaman pengguna yang konsisten di berbagai perangkat (responsif). Styling juga mencakup posisi ikon di dalam *input field* dan transisi warna dasar untuk memberikan umpan balik visual saat pengguna berinteraksi dengan formulir.

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

body {

    font-family: 'Poppins', Arial, sans-serif;
    background-color: #eef2f3;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    margin: 0;
    padding: 20px;
    box-sizing: border-box;
}

.container {
    background-color: white;
    padding: 35px;
    border-radius: 16px;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
    border: 1px solid #e0e0e0;
    width: 100%;
    max-width: 550px;
}
```

```
h1, h2 {  
    text-align: center;  
    color: #1d3557;  
    margin-top: 0;  
    margin-bottom: 30px;  
    font-weight: 600;  
}  
  
.form-group {  
    margin-bottom: 20px;  
}  
  
label {  
    display: block;  
    margin-bottom: 8px;  
    font-weight: 500;  
    color: #457b9d;  
}  
  
.input-with-icon {  
    position: relative;  
}  
  
.input-with-icon svg {  
    position: absolute;  
    top: 50%;  
    left: 15px;  
    transform: translateY(-50%);  
    color: #a8dadc;  
    transition: color 0.3s;
```

```
}

/* Penyesuaian posisi ikon khusus untuk textarea */
.input-with-icon textarea ~ svg {
    top: 22px;
    transform: none;
}

input[type="text"],
input[type="email"],
textarea {
    width: 100%;
    /* Padding kiri tetap besar untuk memberi ruang bagi ikon */
    padding: 15px 15px 15px 50px;
    border: 1px solid #dfe6e9;
    border-radius: 10px;
    box-sizing: border-box;
    transition: all 0.3s;
    background-color: #f8f9fa;
    color: #1d3557;
}

textarea {
    resize: vertical;
    min-height: 120px;
}

/* Efek 'focus' dipertahankan agar interaksi terasa sama */
input:focus,
textarea:focus {
```

```
border-color: #457b9d;

box-shadow: 0 0 0 3px rgba(69, 123, 157, 0.2);

outline: none;
}

input:focus ~ svg,
textarea:focus ~ svg {
    color: #457b9d;
}

button {
    width: 100%;
    padding: 15px;

    /* Gradien disederhanakan menjadi warna solid */
    background-color: #1d3557;
    color: white;
    border: none;
    border-radius: 10px;
    font-size: 16px;
    font-weight: 600;
    cursor: pointer;

    transition: background-color 0.3s ease; /* Transisi
disederhanakan */
}

/* Efek hover dipertahankan dengan mengubah warna latar */
button:hover {
    background-color: #457b9d;
}
```

```
button:disabled {  
    background-color: #a8dadac;  
    cursor: not-allowed;  
}  
  
.error-message {  
    color: #e63946;  
    font-size: 14px;  
    margin-top: 8px;  
    display: none;  
}  
  
.result-card {  
    background-color: #f1faee;  
    border-left: 5px solid #2a9d8f;  
    padding: 25px;  
    margin-bottom: 20px;  
    border-radius: 10px;  
}  
  
hr {  
    border: none;  
    border-top: 1px solid #e0e0e0;  
    margin: 30px 0;  
}
```

4.2.3 script.js (Validasi dan Interaksi jQuery)

File ini mengelola validasi formulir di sisi klien menggunakan jQuery. Skrip akan berjalan ketika pengguna menekan tombol "Kirim Pesan". Prosesnya adalah sebagai berikut: pertama, semua pesan error yang mungkin masih tampil akan disembunyikan menggunakan fungsi `.hide()`. Kedua, skrip akan memeriksa setiap *input* berdasarkan aturan yang ditetapkan: nama dan email tidak boleh kosong, format email harus valid, dan pesan harus lebih dari 10 karakter. Jika ada aturan yang tidak terpenuhi, pesan error yang relevan akan ditampilkan secara instan menggunakan `.show()`. Jika semua input valid, tombol kirim akan dinonaktifkan dan teksnya diubah menjadi "Mengirim..." sebelum formulir dikirim ke server.

```
$(document).ready(function() {

    $('#contactForm').on('submit', function(event) {

        event.preventDefault();

        $('.error-message').hide();

        let isValid = true;

        const nama = $('#nama').val().trim();
        const email = $('#email').val().trim();
        const pesan = $('#pesan').val().trim();
        const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

        // Validasi nama
        if (nama === '') {
            $('#namaError').text('Nama tidak boleh kosong.').show();
            isValid = false;
        }

        // Validasi email
        if (email === '' || !emailRegex.test(email)) {
            $('#emailError').text('Email tidak valid.').show();
            isValid = false;
        }

        // Validasi pesan
        if (pesan.length < 10) {
            $('#pesanError').text('Pesan harus lebih dari 10 karakter.').show();
            isValid = false;
        }

        if (isValid) {
            $('#submit').prop('disabled', true).text('Mengirim...');
            // Di sini Anda bisa menambahkan logika untuk mengirim data ke server
        }
    });
});
```

```
        $('#emailError').text('Email tidak boleh  
kosong.').show();  
  
        isValid = false;  
  
        } else if (!emailRegex.test(email)) {  
  
            $('#emailError').text('Format email tidak  
valid.').show();  
  
            isValid = false;  
  
        }  
  
        if (pesan.length <= 10) {  
  
            $('#pesanError').text('Pesan harus lebih dari 10  
karakter.').show();  
  
            isValid = false;  
  
        }  
  
        if (isValid) {  
  
            const submitButton =  
$(this).find('button[type="submit"]');  
  
            submitButton.prop('disabled',  
true).find('span').text('Mengirim...');  
  
            this.submit();  
  
        }  
  
    });  
  
});
```

4.2.4 proses.php (Logika Backend)

File proses.php berfungsi sebagai *backend* yang menangani data setelah formulir dikirim menggunakan metode POST. Pertama, skrip ini memulai sesi (`session_start()`) untuk menyimpan riwayat pesan. Kemudian, ia mengambil data nama, email, dan pesan dari formulir. Untuk keamanan, data tersebut dibersihkan menggunakan `htmlspecialchars()` untuk mencegah serangan XSS. Data yang telah dibersihkan beserta waktu pengiriman disimpan ke dalam sebuah *array* di dalam `$_SESSION['messages']`. Terakhir, halaman ini akan menampilkan halaman konfirmasi "Pesan Berhasil Terkirim!", data yang baru saja dikirim oleh pengguna, serta seluruh riwayat pesan yang tersimpan di sesi. Halaman ini juga menyertakan sedikit jQuery untuk memberikan efek animasi *fade-in* pada kartu hasil.

```
<?php

// Memulai atau melanjutkan sesi yang sudah ada
session_start();

// Mengatur zona waktu default ke Waktu Indonesia Barat
date_default_timezone_set('Asia/Jakarta');

// Cek apakah variabel session 'messages' sudah ada. Jika tidak,
// buat sebagai array kosong.
if (!isset($_SESSION['messages'])) {
    $_SESSION['messages'] = [];
}

// Inisialisasi variabel untuk menampung data dari form
$nama = $email = $pesan = '';

// Cek apakah permintaan ke halaman ini menggunakan metode POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Ambil data dari form dan bersihkan dari tag HTML untuk
    // keamanan
    $nama = htmlspecialchars($_POST['nama']);
    $email = htmlspecialchars($_POST['email']);
```



```

$pesan = htmlspecialchars($_POST['pesan']);

// Buat array baru untuk menyimpan data pesan yang baru dikirim
$newMessage = [
    'nama' => $nama,
    'email' => $email,
    'pesan' => $pesan,
    'waktu' => date('d M Y, H:i') // Catat waktu pengiriman
];

// Masukkan pesan baru ke dalam array 'messages' di session
array_push($_SESSION['messages'], $newMessage);

} else {
    // Jika halaman diakses langsung (bukan dari form), alihkan
    kembali ke index.html

    header('Location: index.html');

    exit(); // Hentikan eksekusi skrip
}

?>

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Status Pengiriman</title>

    <link rel="stylesheet" href="style.css">
</head>
<body>

```

```

<div class="container" style="text-align: center;">

    <h1>Pesan Berhasil Terkirim!</h1>

    <div class="result-card" style="opacity:0; text-align:
left;">

        <p>Terima kasih, kami telah menerima pesan Anda.</p>

        <hr style="margin: 20px 0;">

        <p><strong>Data yang Anda Kirim:</strong></p>

        <p>

            <strong>Nama:</strong> <?php echo $nama; ?><br>

            <strong>Email:</strong> <?php echo $email; ?><br>

            <strong>Pesan:</strong> <?php echo nl2br($pesan); ?>

        </p>
    </div>

    <a href="index.html" style="margin-bottom: 30px; display:
inline-block;">Kirim Pesan Lain</a>

    <hr>

    <h2>Riwayat Pesan</h2>

    <div class="history-container">

        <?php if (!empty($_SESSION['messages'])): ?>

            <?php foreach (array_reverse($_SESSION['messages'])
as $msg): ?>

                <div class="result-card" style="text-align:
left; background-color: #f8f9fa; border-left-color: #6c757d;">

                    <p style="font-size: 14px; color: #6c757d;">

                        <em>Dikirim pada: <?php echo
$msg['waktu']; ?></em>

                    </p>

                    <p>

```

```
                <strong>Nama:</strong> <?php echo
htmlspecialchars($msg['nama']); ?><br>

                <strong>Email:</strong> <?php echo
htmlspecialchars($msg['email']); ?><br>

                <strong>Pesan:</strong> <?php echo
nl2br(htmlspecialchars($msg['pesan'])); ?>

            </p>

        </div>

        <?php endforeach; ?>

    <?php endif; ?>

</div>

</div>

<script src="jquery-3.7.1.js"></script>

<script>

    $(document).ready(function() {

        // Animasi untuk memunculkan kartu hasil yang pertama
        $('.result-card').first().animate({'opacity': '1'},
800);

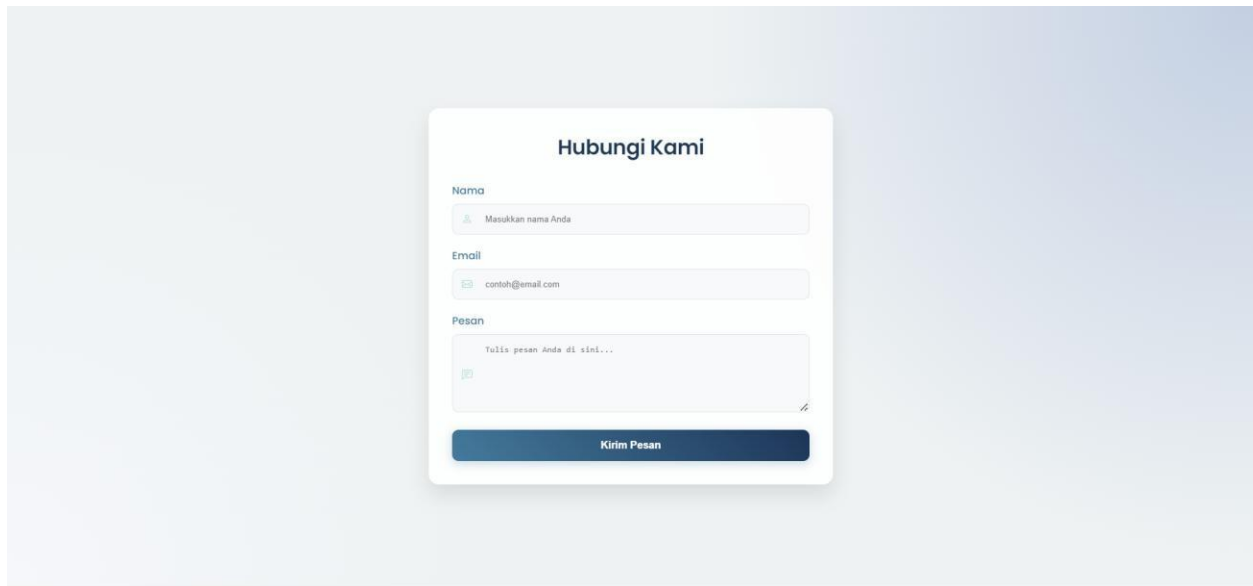
    });

</script>

</body>

</html>
```

Output



Hubungi Kami

Nama

Masukkan nama Anda

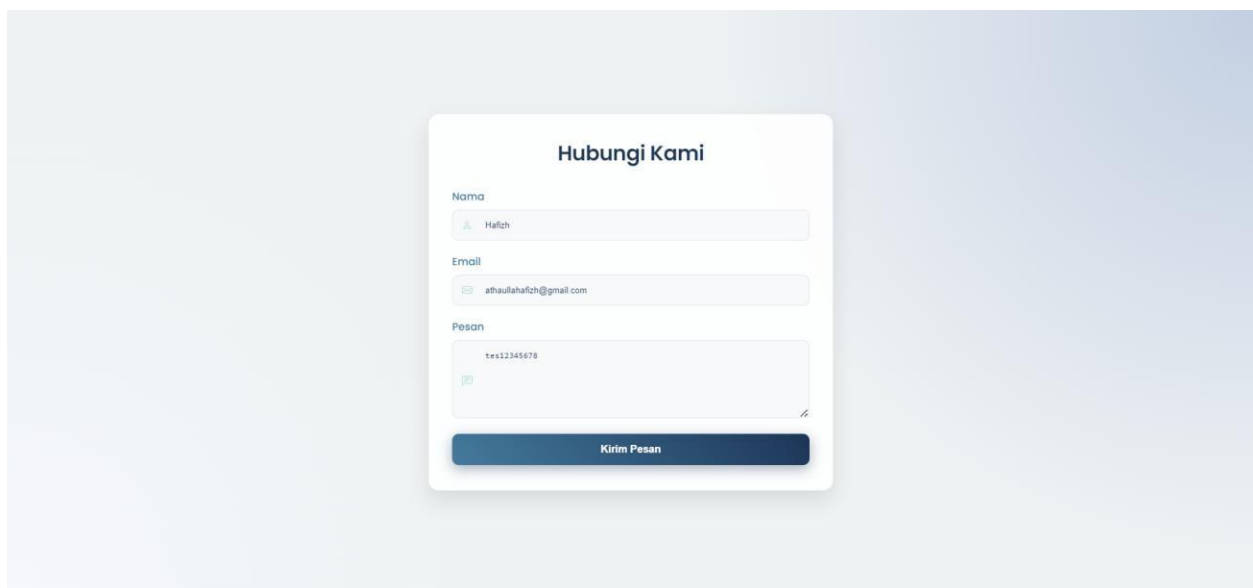
Email

cantoh@email.com

Pesan

Tulis pesan Anda di sini...

Kirim Pesan



Hubungi Kami

Nama

Haftah

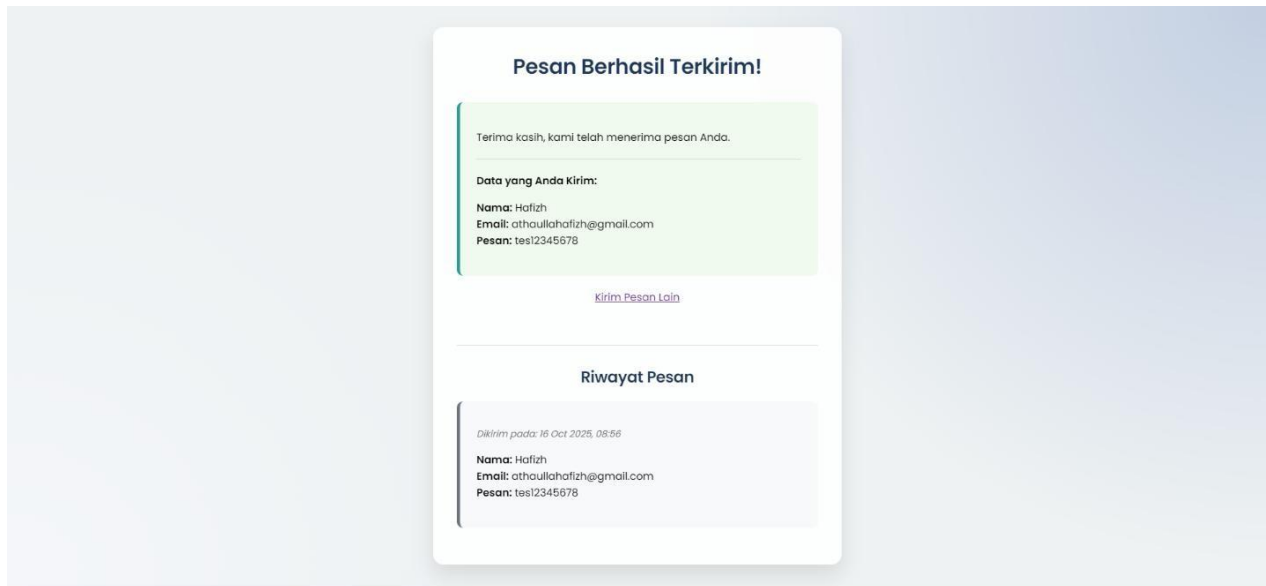
Email

athaulahafizh@gmail.com

Pesan

tes12345678

Kirim Pesan



VII. KESIMPULAN

7.1 KESIMPULAN

Proyek "Sistem Kontak Pengguna Interaktif" telah berhasil dikembangkan dan diimplementasikan sesuai dengan tujuan yang telah ditetapkan. Aplikasi ini berhasil menyediakan sebuah formulir kontak yang fungsional, interaktif, dan menarik secara visual. Semua kriteria penugasan utama telah terpenuhi, termasuk pembuatan formulir HTML yang terstruktur, validasi input di sisi klien dengan jQuery, pemrosesan data di sisi server menggunakan PHP, serta penerapan desain yang responsif dan modern dengan CSS. Penggunaan animasi jQuery juga berhasil meningkatkan pengalaman pengguna, membuat interaksi terasa lebih dinamis dan modern.

7.2 REKOMENDASI

Untuk pengembangan proyek di masa mendatang, beberapa peningkatan dapat dipertimbangkan:

- **Implementasi AJAX:** Mengimplementasikan fitur opsional pengiriman data menggunakan AJAX dengan jQuery. Ini akan memungkinkan formulir dikirim tanpa me-reload halaman, memberikan pengalaman pengguna yang lebih mulus dan cepat.
- **Penyimpanan Data Permanen:** Mengganti sistem simulasi dengan koneksi ke database (seperti MySQL) untuk menyimpan pesan dari pengguna secara permanen, bukan hanya menampilkannya sesaat.
- **Fungsi Pengiriman Email:** Mengintegrasikan library seperti PHPMailer untuk benar-benar mengirimkan pesan yang diisi pengguna ke alamat email administrator secara otomatis.

- **Keamanan Lanjutan:** Menambahkan mekanisme keamanan seperti Google reCAPTCHA untuk melindungi formulir dari pengiriman spam otomatis oleh bot.

VIII. DAFTAR PUSTAKA

Daftar referensi yang digunakan selama pengembangan proyek ini:

- Dokumentasi Resmi PHP. Diakses dari <https://www.php.net/manual/en/>.
- Dokumentasi API jQuery. Diakses dari <https://api.jquery.com/>.
- MDN Web Docs (Mozilla). Diakses dari <https://developer.mozilla.org/>.
- W3Schools Online Web Tutorials. Diakses dari <https://www.w3schools.com/>.

IX. REVISI

Menggunakan metode penyimpanan JSON, perubahan pada kode php :

```
<?php

date_default_timezone_set('Asia/Jakarta');

$filePenyimpanan = 'riwayat_pesan.json';

// Fungsi untuk membaca semua pesan dari file
function getMessages($file) {
    if (!file_exists($file)) {
        return [];
    }

    $data = file_get_contents($file);
    $messages = json_decode($data, true);

    // Jika file kosong atau formatnya salah, kembalikan array
    kosong

    return $messages === null ? [] : $messages;
}
```

```
// Inisialisasi variabel

$nama = $email = $pesan = '';

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Ambil data dari form

    $nama = htmlspecialchars($_POST['nama']);
    $email = htmlspecialchars($_POST['email']);
    $pesan = htmlspecialchars($_POST['pesan']);

    // Baca data yang sudah ada

    $messages = getMessages($filePenyimpanan);

    // Buat array untuk pesan baru

    $newMessage = [

        'nama' => $nama,
        'email' => $email,
        'pesan' => $pesan,
        'waktu' => date('d M Y, H:i')

    ];

    array_push($messages, $newMessage);

    file_put_contents($filePenyimpanan, json_encode($messages,
JSON_PRETTY_PRINT));

} else {

    header('Location: index.html');

    exit();

}
```

```
?>

<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Status Pengiriman</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container" style="text-align: center;">

        <h1>Pesan Berhasil Terkirim!</h1>

        <div class="result-card" style="opacity:0; text-align:
left;">

            <p>Terima kasih, kami telah menerima pesan Anda.</p>

            <hr style="margin: 20px 0;">

            <p><strong>Data yang Anda Kirim:</strong></p>

            <p>

                <strong>Nama:</strong> <?php echo $nama; ?><br>

                <strong>Email:</strong> <?php echo $email; ?><br>

                <strong>Pesan:</strong> <?php echo nl2br($pesan); ?>

            </p>

        </div>

        <a href="index.html" style="margin-bottom: 30px; display:
inline-block;">Kirim Pesan Lain</a>

    </div>

</body>

</html>
```



```
<hr>

<h2>Riwayat Pesan</h2>

<div class="history-container">

    <?php

        // Baca semua pesan untuk ditampilkan

        $semuaPesan = getMessages($filePenyimpanan);

        if (!empty($semuaPesan)):

            ?>

                <?php foreach (array_reverse($semuaPesan) as $msg): ?>

                    <div class="result-card" style="text-align: left;
background-color: #f8f9fa; border-left-color: #6c757d;">

                        <p style="font-size: 14px; color: #6c757d;">

                            <em>Dikirim pada: <?php echo $msg['waktu'];

?></em>

                                </p>

                                <p>

                                    <strong>Nama:</strong> <?php echo
htmlspecialchars($msg['nama']); ?><br>

                                    <strong>Email:</strong> <?php echo
htmlspecialchars($msg['email']); ?><br>

                                    <strong>Pesan:</strong> <?php echo
nl2br(htmlspecialchars($msg['pesan'])); ?>

                                </p>

                            </div>

                        <?php endforeach; ?>

                    <?php endif; ?>

                </div>

            </div>

            <script src="jquery-3.7.1.js"></script>
```

```
<script>

    $(document).ready(function() {

        $('.result-card').first().animate({'opacity': '1'}, 800);

    });

</script>

</body>

</html>
```

Dan penambahan file JSON

 riwayat_pesan.json