

Laporan Praktikum
Mata Kuliah Pemrograman Web



Pertemuan 5.Praktikum3
“Pembuatan Website Sederhana dengan Node.js dan HTML”

Dosen Pengampu :
Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh :
Siti Nurkholizah

2308807

3A - Sistem Informasi Kelautan

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi informasi, pembuatan aplikasi web menjadi semakin mudah, salah satunya dengan memanfaatkan Node.js. Node.js memungkinkan penggunaan JavaScript di sisi server, sehingga pengembang dapat menggunakan satu bahasa pemrograman di kedua sisi (klien dan server).

Dalam pengembangan web, HTML digunakan untuk membangun struktur halaman, sementara CSS berfungsi untuk mengatur tampilan visual agar lebih menarik dan responsif. Pada praktikum ini, kita akan belajar membuat server sederhana menggunakan Node.js. Server tersebut akan menampilkan halaman web statis yang dihasilkan dari file HTML dan CSS. Dengan memanfaatkan modul “http” dan “fs” di Node.js, kita akan mengatur server lokal yang membaca file dari sistem dan menampilkannya kepada pengguna melalui browser. Praktikum ini akan membantu memahami konsep dasar pembuatan server web sederhana dan komunikasi antara klien dan server melalui protokol HTTP.

II. TUJUAN

Tujuan dari praktikum ini adalah untuk mempelajari cara membuat server web sederhana menggunakan Node.js. Selain itu, praktikum ini bertujuan untuk memahami proses pengiriman halaman HTML dari server ke browser pengguna, serta mempraktikkan penggunaan modul “http” dan “fs” pada Node.js untuk membaca file dari sistem. Akhirnya, kami juga bertujuan untuk membuat dan menampilkan halaman web statis yang menarik dengan memanfaatkan HTML dan CSS.

III. ALAT DAN BAHAN

- Node.js yang sudah terinstall di komputer.
- Visual Studio Code
- Browser web (Google Chrome)
- Terminal atau Command Prompt.
- File HTML dan CSS.

IV. LANGKAH KERJA

1. Membuat Server dengan node.js

Langkah pertama yaitu membuat file bernama server.js. Didalam file ini dapat memanfaatkan modul http dan fs dari Node.js untuk membuat server yang membaca dan menyajikan file HTML

Source code dibuat seperti dibawah ini

```
const http = require('http');
const fs = require('fs');

const server = http.createServer((req, res) => {
  fs.readFile('index.html', function(err, data){
    res.statusCode = 200; // server berjalan dengan baik
    res.setHeader('Content-Type', 'text/html');
    res.write(data);
    return res.end();
  });
});

server.listen(3000, () => {
  console.log("Server Berhasil Berjalan di Port 3000");
});
```

2. Membuat Halaman Web

Setelah itu buat file HTML yang bernama index.html dan CSS yang sesuai dengan struktur website.

Source code index.html dibuat seperti ini untuk menampilkan halaman sederhana seperti header, content, dan footer.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Website Sederhana</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    body {
      background-color: #fdd0e8;
      font-family: sans-serif;
    }

    .content {
      grid-area: content;
```

```
}
```

```
.sidebar {  
  grid-area: sidebar;  
  background: linear-gradient(  
    to right,  
    rgb(245, 79, 143),  
    rgb(247, 105, 250),  
    rgba(110, 125, 253, 1)  
  );  
  justify-content: center;  
}
```

```
.footer {  
  grid-area: footer;  
  background: #ee8ead;  
}
```

```
.container {  
  font-size: 1.5em;  
  width: 100%;  
  height: 100;  
  height: 100vh;  
  display: grid;  
  grid-template-areas: 'sidebar' 'content' 'footer';  
  grid-template-columns: 1fr;  
  grid-template-rows: 130px 800px 250px;  
}
```

```
.content,  
.sidebar,  
.footer {  
  padding: 1em;  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  display: flex;  
  justify-content: space-between;  
  text-align: center;  
}
```

```
nav li {  
  list-style: none;  
  padding: 1em 0;  
}
```

```
nav li a {  
  color: rgb(31, 98, 252);;  
  font-weight: 700;  
  opacity: 0.6;  
  text-decoration: none;  
  transition: 0.3s;  
}
```

```
nav li a:hover {  
  opacity: 1;  
}
```

```
.hero {  
  max-width: 90 px;  
  margin: 0 auto;  
  text-align: center;  
}
```

```
.hero img {  
  width: 200px;  
}
```

```
.hero h1 {  
  font-size: 2em;  
  font-weight: 300;  
  color: rgb(255, 121, 190);;  
}
```

```
.hero h2 {  
  font-size: 2;  
  font-weight: 100;  
  color: rgb(255, 121, 190);;  
}
```

```
.hero p {  
  font-weight: 300;  
  line-height: 1.3em;  
  color: rgb(31, 98, 252);;
```

```
}
```

```
.action-btn {  
  display: inline-block;  
  text-decoration: none;  
  color: rgb(255, 181, 222);  
  font-weight: 700;  
  background: #fb568a;  
  padding: 0.5em 2em;  
  border-radius: 60px;  
  margin: 1em 0;  
  transition: 0.3s;  
}
```

```
footer ul {  
  max-width: 640px;  
  margin: 2em auto;  
  padding: 0;  
  text-align: center;  
  display: flex;  
  flex-direction: row;  
}
```

```
footer ul li {  
  list-style: none;  
  align-self: flex-end;  
}
```

```
footer ul li a {  
  text-decoration: none;  
  color: #335fa5;  
}
```

```
footer ul li img {  
  width: 30%;  
}
```

```
footer p {  
  font-size: 0.8em;  
}
```

```
@media (min-width: 1040px) {  
  .container {
```

```
    grid-template-areas: 'sidebar content' 'sidebar footer';
    grid-template-rows: 1fr auto;
    grid-template-columns: 300px 1f;
  }
```

```
nav ul {
  display: flex;
  justify-content: space-between;
  flex-direction: column;
}
```

```
.sidebar {
  background: linear-gradient(
    rgb(255, 190, 215),
    rgb(168, 203, 255),
    rgb(177, 144, 255)
  );
  padding-top: 1em;
}
```

```
.hero {
  text-align: left;
  margin: 1em 0;
}
```

```
.hero img {
  width: 100px;
  float: top;
}
```

```
.hero h1 {
  font-size: 3em;
}
```

```
.hero p {
  width: 60%;
}
```

```
footer ul {
  max-width: 900px;
  margin: 0 auto;
  padding: 1em 0;
}
```

```

        footer ul li a img {
            width: 50%;
        }
    }
</style>
</head>
<body>

```

```

<div class="container">
    <div class="sidebar">
        <nav>
            <ul>
                <li><a href="about.html">About</a></li>
                <li><a href="porto.html">Portofolio</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
        </nav>
    </div>
    <main class="content">
        <section class="hero">
            
            <div class="hero-content">
                <h1>Profesi</h1>
                <br>
                <h2>Junior Content Writer at Dicoding</h2>
                <br>
                <p style="text-align: justify;">

```

Seorang Junior Content Writer juga berkontribusi dalam pembuatan materi kursus online seperti modul, kuis, dan tugas akhir. Riset tentang tren teknologi terbaru menjadi bagian penting dari pekerjaan ini,

sehingga konten yang disajikan selalu relevan dan up-to-date. Posisi ini juga melibatkan kolaborasi dengan tim pengembang, desainer, dan produk untuk memastikan kualitas dan konsistensi materi.

Dengan kemampuan menulis yang baik dan pengetahuan dasar tentang teknologi, profesi ini memberi peluang untuk terus belajar dan berkembang dalam dunia teknologi sambil berkontribusi dalam edukasi digital di Indonesia.

```

            </p>
            <a href="pp.html" class="action-btn">Profile Saya</a>
        </div>

```



```

        </section>
    </main>
    <div class="footer">
        <footer>
            <ul>
                <li>
                    <a><p>Instagram</p></a>
                </li>
                <li>
                    <a><p>Facebook</p></a>
                </li>
                <li>
                    <a><p>Twitter</p></a>
                </li>
                <li>
                    <a><p>Telegram</p></a>
                </li>
            </ul>
        </footer>
    </div>
</div>
</body>
</html>

```

```

About.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Website Sederhana</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        body {

```

```
background-color: #fdd0e8;
font-family: sans-serif;
}

.content {
  grid-area: content;
}

.sidebar {
  grid-area: sidebar;
  background: linear-gradient(
    to right,
    rgb(245, 79, 143),
    rgb(247, 105, 250),
    rgba(110, 125, 253, 1)
  );
  justify-content: center;
}

.footer {
  grid-area: footer;
  background: #ee8ead;
}

.container {
  font-size: 1.5em;
  width: 100%;
  height: 100;
  height: 100vh;
  display: grid;
  grid-template-areas: 'sidebar' 'content' 'footer';
  grid-template-columns: 1fr;
  grid-template-rows: 130px 800px 250px;
}

.content,
.sidebar,
.footer {
  padding: 1em;
}

nav ul {
  margin: 0;
```

```
padding: 0;
display: flex;
justify-content: space-between;
text-align: center;
}
```

```
nav li {
  list-style: none;
  padding: 1em 0;
}
```

```
nav li a {
  color: rgb(31, 98, 252);;
  font-weight: 700;
  opacity: 0.6;
  text-decoration: none;
  transition: 0.3s;
}
```

```
nav li a:hover {
  opacity: 1;
}
```

```
.hero {
  max-width: 90 px;
  margin: 0 auto;
  text-align: center;
}
```

```
.hero img {
  width: 200px;
}
```

```
.hero h1 {
  font-size: 2em;
  font-weight: 300;
  color: rgb(255, 121, 190);;
}
.hero h2 {
  font-size: 2;
  font-weight: 100;
  color: rgb(255, 121, 190);;
}
```

```
.hero p {  
  font-weight: 300;  
  line-height: 1.3em;  
  color: rgb(31, 98, 252);  
}
```

```
.action-btn {  
  display: inline-block;  
  text-decoration: none;  
  color: rgb(255, 181, 222);  
  font-weight: 700;  
  background: #fb568a;  
  padding: 0.5em 2em;  
  border-radius: 60px;  
  margin: 1em 0;  
  transition: 0.3s;  
}
```

```
footer ul {  
  max-width: 640px;  
  margin: 2em auto;  
  padding: 0;  
  text-align: center;  
  display: flex;  
  flex-direction: row;  
}
```

```
footer ul li {  
  list-style: none;  
  align-self: flex-end;  
}
```

```
footer ul li a {  
  text-decoration: none;  
  color: #335fa5;  
}
```

```
footer ul li img {  
  width: 30%;  
}
```

```
footer p {  
  font-size: 0.8em;  
}
```

```
@media (min-width: 1040px) {  
  .container {  
    grid-template-areas: 'sidebar content' 'sidebar footer';  
    grid-template-rows: 1fr auto;  
    grid-template-columns: 300px 1f;  
  }
```

```
  nav ul {  
    display: flex;  
    justify-content: space-between;  
    flex-direction: column;  
  }
```

```
  .sidebar {  
    background: linear-gradient(  
      rgb(255, 190, 215),  
      rgb(168, 203, 255),  
      rgb(177, 144, 255)  
    );  
    padding-top: 1em;  
  }
```

```
  .hero {  
    text-align: left;  
    margin: 1em 0;  
  }
```

```
  .hero img {  
    width: 100px;  
    float: top;  
  }
```

```
  .hero h1 {  
    font-size: 3em;  
  }
```

```
  .hero p {  
    width: 60%;  
  }
```

```

    footer ul {
        max-width: 900px;
        margin: 0 auto;
        padding: 1em 0;
    }

    footer ul li a img {
        width: 50%;
    }
}
</style>
</head>
<body>

<div class="container">
    <div class="sidebar">
        <nav>
            <ul>
                <li><a href="about.html">About</a></li>
                <li><a href="porto.html">Portofolio</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
        </nav>
    </div>
    <main class="content">
        <section class="hero">
            
            <div class="hero-content">
                <h1>Profesi</h1>
                <br>
                <h2>Junior Content Writer at Dicoding</h2>
                <br>
                <p style="text-align: justify;">

```

Seorang Junior Content Writer juga berkontribusi dalam pembuatan materi kursus online seperti modul, kuis, dan tugas akhir. Riset tentang tren teknologi terbaru menjadi bagian penting dari pekerjaan ini,

sehingga konten yang disajikan selalu relevan dan up-to-date. Posisi ini juga melibatkan kolaborasi dengan tim pengembang, desainer, dan produk untuk memastikan kualitas dan konsistensi materi.

Dengan kemampuan menulis yang baik dan pengetahuan dasar tentang teknologi, profesi ini memberi peluang untuk terus belajar dan berkembang dalam dunia teknologi sambil berkontribusi dalam edukasi digital di Indonesia.

```
</p>
<a href="pp.html" class="action-btn">Profile Saya</a>
</div>
</section>
</main>
<div class="footer">
<footer>
<ul>
<li>
<a><p>Instagram</p></a>
</li>
<li>
<a><p>Facebook</p></a>
</li>
<li>
<a><p>Twitter</p></a>
</li>
<li>
<a><p>Telegram</p></a>
</li>
</ul>
</footer>
</div>
</div>
</body>
</html>
```

Blog.html

```
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Blog</title>
```

```

    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <header>
            <h1>Blog</h1>
        </header>
        <section class="blog-content">
            <article>
                <h2>Cara Meningkatkan Performa Web</h2>
                <p>Performa web sangat penting untuk pengalaman pengguna
dan peringkat SEO. Pelajari cara mengoptimalkan gambar, mengurangi
waktu muat halaman, dan meningkatkan kecepatan situs web secara
keseluruhan.</p>
            </article>
            <article>
                <h2>Desain Web Responsif</h2>
                <p>Desain responsif adalah kunci untuk memastikan situs
web Anda terlihat bagus di semua perangkat. Temukan praktik terbaik
dan alat untuk membuat situs web yang ramah seluler. </p>
            </article>
        </section>
        <footer>
            <a href="index.html">Kembali ke Beranda</a>
        </footer>
    </div>
</body>
</html>

```

Contact.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Contact</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <header>

```



```

        <h1>Contact Me</h1>
    </header>
    <section class="contact-content">
        <form action="#" method="POST">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>

            <label for="message">Message:</label>
            <textarea id="message" name="message" rows="5"
required></textarea>

            <button type="submit">Send Message</button>
        </form>
    </section>
    <footer>
        <a href="index.html">Back to Home</a>
    </footer>
</div>
</body>
</html>

```

Profil.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Profil Saya</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f75872;
            margin: 0;
            padding: 0;
        }

        .container {
            width: 80%;

```

```
    margin: 0 auto;
    padding: 20px;
    background-color: #b2cef9;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
.profile-header {
    text-align: center;
    padding: 20px;
    background-color: #f8608e;
    color: #0a38de;
    margin-bottom: 20px;
}
```

```
.profile-header h1 {
    margin: 0;
    font-size: 2.5em;
}
```

```
.profile-header p {
    margin: 5px 0;
    font-size: 1.2em;
}
```

```
.profile-content {
    display: flex;
    flex-direction: column;
    align-items: center;
}
```

```
.profile-content img {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    margin-bottom: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}
```

```
.profile-content h2 {
    color: #0a38de;
    font-size: 2em;
    margin-bottom: 10px;
}
```

```

.profile-content p {
    color: #0a38de;
    font-size: 1.1em;
    line-height: 1.6;
    text-align: center;
    max-width: 600px;
}

footer {
    text-align: center;
    margin-top: 30px;
}

footer a {
    text-decoration: none;
    color: #f8608e;
    font-weight: bold;
}

footer a:hover {
    color: #f8608e;
}
</style>
</head>
<body>

<div class="container">
    <div class="profile-header">
        <h1>Profil Saya</h1>
        <p>Web Developer & Content Creator</p>
    </div>

    <div class="profile-content">
        
        <h2>Siti Nurkholizah</h2>
        <p style="text-align: justify;">
            Saya seorang pengembang web dengan pengalaman dalam
            membangun website menggunakan teknologi seperti HTML, CSS,
            JavaScript, dan framework modern. Saya bersemangat dalam
            menciptakan aplikasi web yang fungsional dan menarik secara visual.
        </p>
    </div>

```

```

        <footer>
            <a href="index.html">Kembali ke Beranda</a>
        </footer>
    </div>

</body>
</html>

```

Portfolio.html

```

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Portofolio</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <header>
            <h1>Portofolio</h1>
        </header>
        <section class="portfolio-content">
            <ul>
                <li><strong>Proyek 1:</strong> Situs e-commerce yang
dibangun dengan HTML, CSS, dan JavaScript</li>
                <li><strong>Proyek 2:</strong> Platform blog yang
dikembangkan menggunakan React.js dan Node.js</li>
                <li><strong>Proyek 3:</strong> Situs perusahaan yang
menggunakan Bootstrap dan jQuery</li>
            </ul>
        </section>
        <footer>
            <a href="index.html">Kembali ke Beranda</a>
        </footer>
    </div>
</body>
</html>

```

3. Membuat gaya visual dengan CSS

Menambahkan file CSS untuk memberikan tampilan yang menarik pada halaman web.

Style.css

```
/* Global Styles */
```

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  background-color: #f75872;  
}
```

```
.container {  
  width: 80%;  
  margin: 0 auto;  
  padding: 20px;  
  background-color: #b2cef9;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}
```

```
header h1 {  
  text-align: center;  
  color: #0a38de;  
  font-size: 2.5em;  
  margin-bottom: 20px;  
}
```

```
footer {  
  text-align: center;  
  margin-top: 30px;  
}
```

```
footer a {  
  text-decoration: none;  
  color: #0a38de;  
  font-weight: bold;  
}
```

```
footer a:hover {  
  color: #0a38de;  
}
```

```
/* About Page */
.about-content p {
    text-align: justify;
    line-height: 1.6;
}

/* Portfolio Page */
.portfolio-content ul {
    list-style: none;
    padding: 0;
}

.portfolio-content ul li {
    background-color: #87b1ec;
    margin-bottom: 10px;
    padding: 10px;
    border-left: 5px solid #567bfb;
}

/* Blog Page */
.blog-content article {
    margin-bottom: 20px;
}

.blog-content article h2 {
    color: #333;
}

.blog-content article p {
    line-height: 1.6;
}

/* Contact Page */
.contact-content form {
    display: flex;
    flex-direction: column;
}

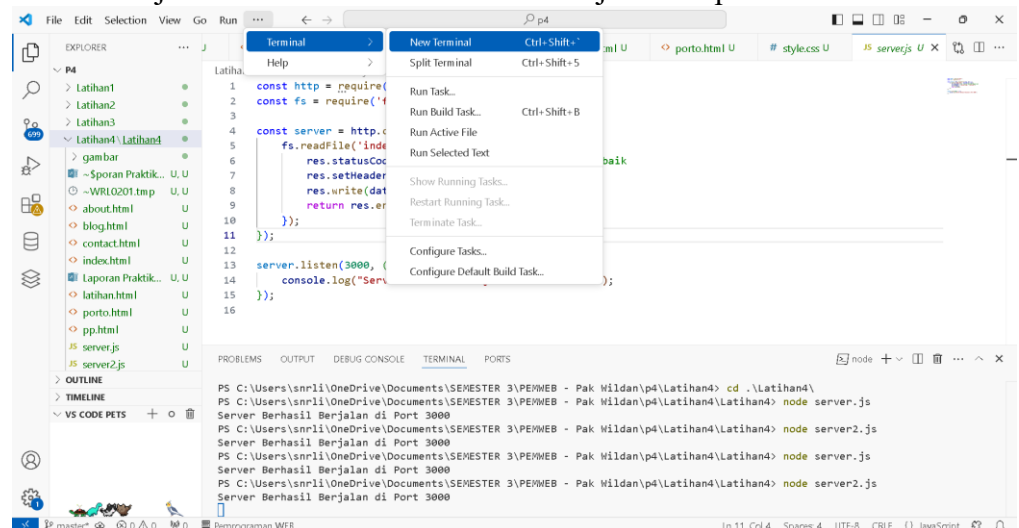
.contact-content form label {
    margin-top: 10px;
    margin-bottom: 5px;
    color: #333;
}
```

```
.contact-content form input,
.contact-content form textarea {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 1em;
}
```

```
.contact-content form button {
  margin-top: 20px;
  padding: 10px;
  background-color: #567bf6;
  color: white;
  border: none;
  border-radius: 5px;
  font-size: 1em;
  cursor: pointer;
}
```

```
.contact-content form button:hover {
  background-color: #3751c1;
}
```

4. Untuk menjalankan server buka terminal dan jalankan perintah berikut

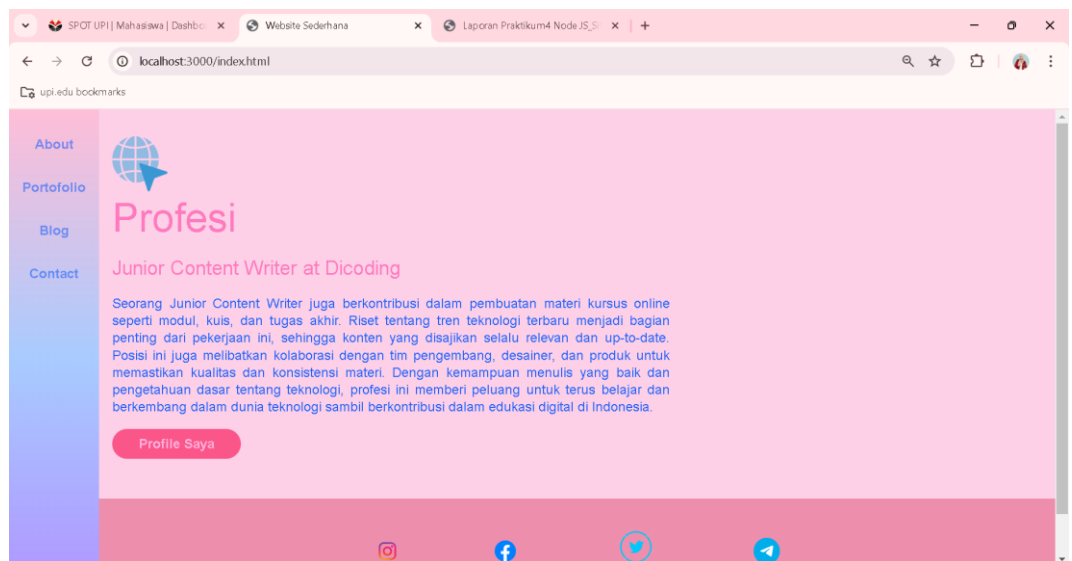


Kemudiann server akan berjalan diport 3000 dan dapat mengakses halaman web di browser melalui URL : <http://localhost:3000>

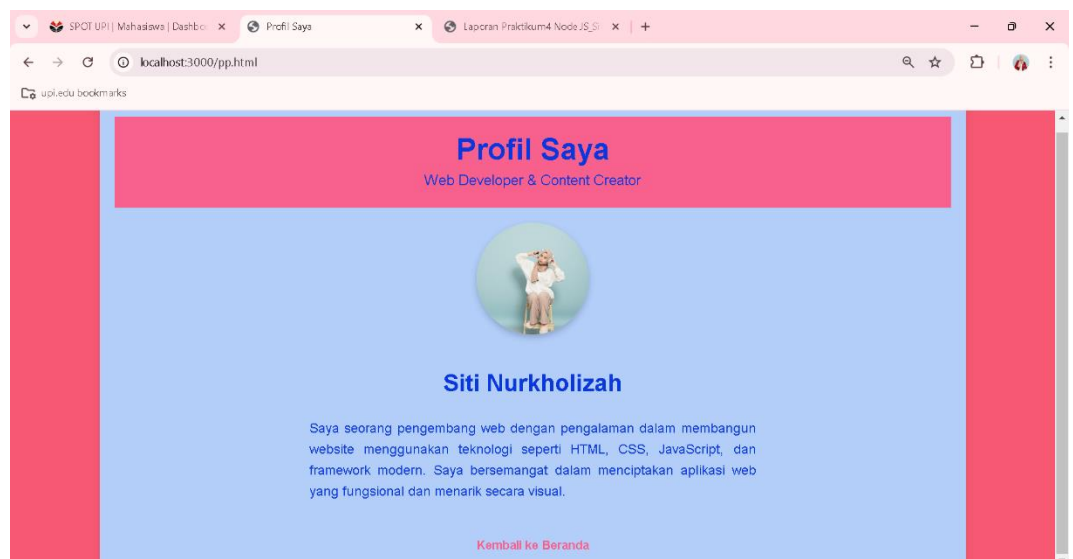
V. HASIL DAN PEMBAHASAN

Setelah server dijalankan, halaman **index.html** akan tampil di browser ketika mengakses <http://localhost:3000>. Proses ini menunjukkan bagaimana server Node.js membaca file dari sistem dan mengirimkannya sebagai tanggapan ke browser. Dalam praktik ini, saya berhasil membuat server sederhana dan memahami bagaimana data diambil dari file sistem serta ditampilkan sebagai halaman web.

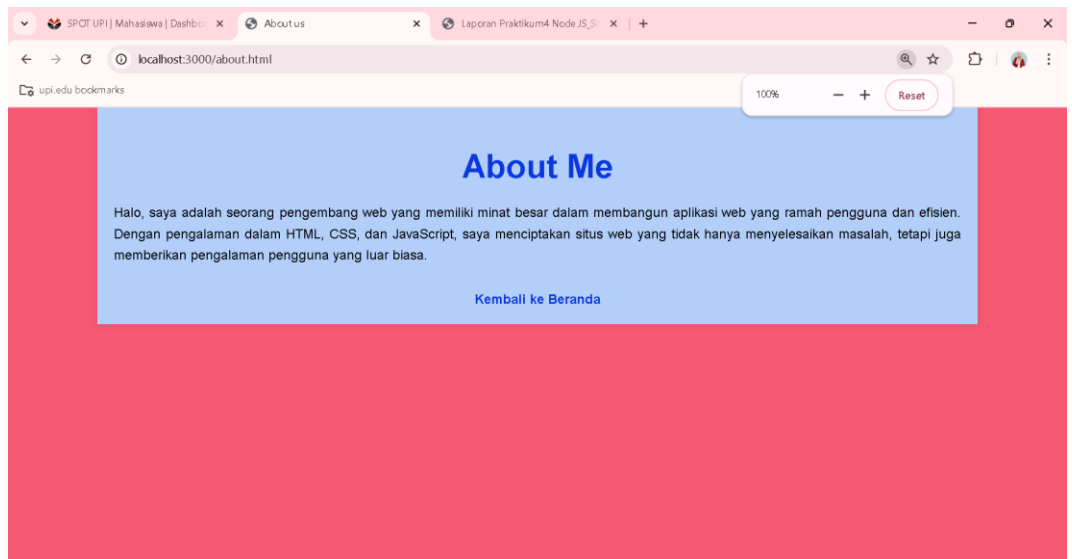
Tampilan index.html



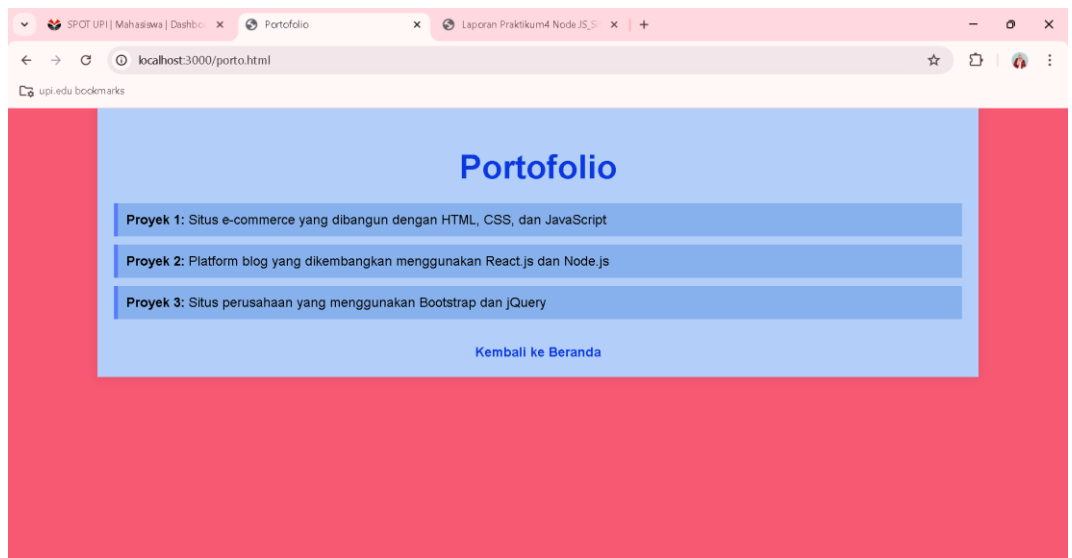
Profil.html



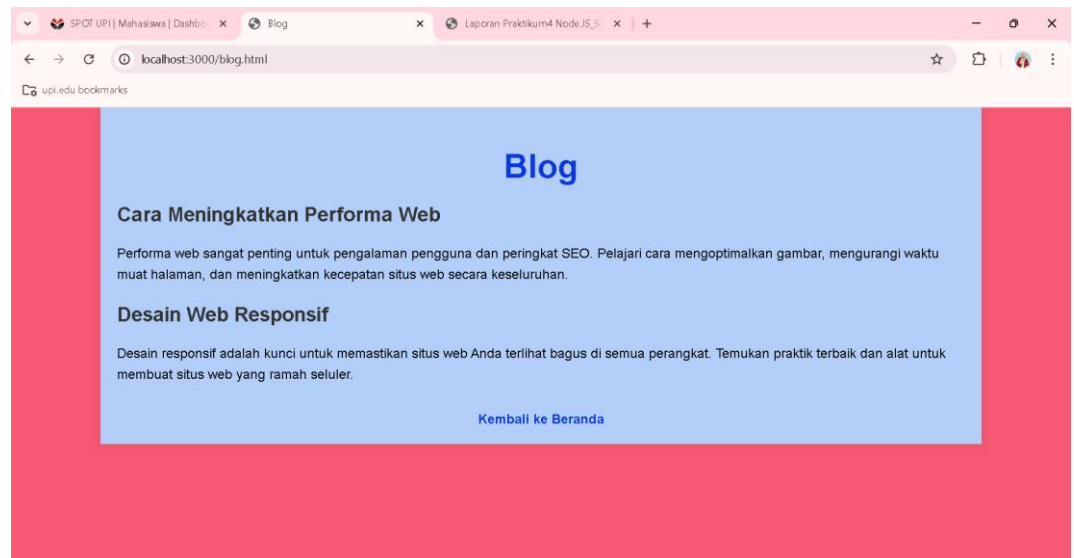
About.html



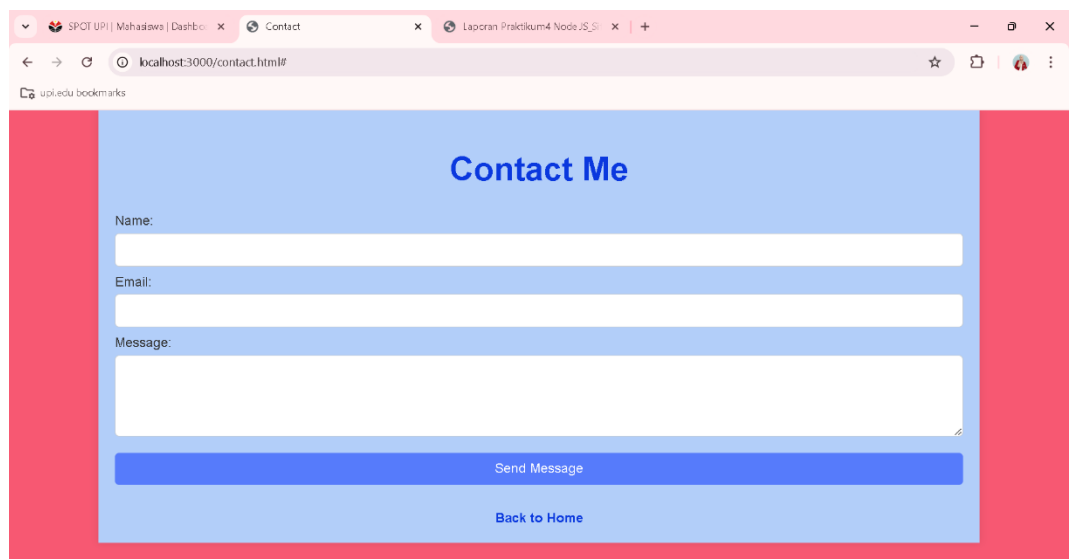
Portofolio.html



Blog.html



Contact.html



VI. KESIMPULAN

Dalam praktikum ini, saya mempelajari cara membuat server web sederhana menggunakan Node.js yang melayani halaman **HTML** dan **CSS**. Dengan memanfaatkan modul **http** dan **fs**, server dapat membaca file dan mengirimkannya ke browser pengguna. Node.js menyediakan cara yang efisien untuk membuat server web tanpa memerlukan banyak alat tambahan. Kami juga memahami bagaimana menghubungkan file **HTML** dengan **CSS** untuk menciptakan tampilan web yang menarik.