

Programmer dan Contoh Mini Project dari Programmer

Shandy Muhammad Yusuf

Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

Article Info

Article history:

Received 2023

Revised 2023

Accepted 2023

Keywords:

Profesionalisme

Programmer

Kode etik programmer

ABSTRACT

Pemrograman berbasis web adalah salah satu bidang yang banyak diminati oleh para programmer, karena memiliki potensi yang besar untuk mengembangkan aplikasi yang interaktif, dinamis, dan mudah diakses. Namun, pemrograman berbasis web juga memiliki tantangan tersendiri, seperti kompleksitas, keamanan, dan keterbatasan sumber daya. Oleh karena itu, diperlukan suatu metode atau pendekatan yang efektif dan efisien untuk membangun aplikasi web yang berkualitas. Penelitian ini bertujuan untuk mengkaji dan membandingkan beberapa metode atau pendekatan pemrograman berbasis web, yaitu waterfall, agile, dan model-driven development. Metode penelitian yang digunakan adalah studi literatur, dengan mengumpulkan, menganalisis, dan menyintesis berbagai sumber yang relevan. Hasil penelitian menunjukkan bahwa setiap metode atau pendekatan memiliki kelebihan dan kekurangan masing-masing, tergantung pada jenis, skala, dan konteks aplikasi web yang dibangun. Penelitian ini diharapkan dapat memberikan wawasan dan rekomendasi bagi para programmer yang ingin memilih metode atau pendekatan yang sesuai untuk proyek pemrograman berbasis web mereka.

Corresponding Author:

Shandy Muhammad Yusuf,

Informatics Department, Faculty of Science & Technology, UIN Sunan Gunung Djati Bandung

Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614

Email: muhammadyusufshandy@gmail.com

1. Pendahuluan

Programmer adalah salah satu profesi yang berkembang pesat di era digital saat ini. Programmer adalah orang yang menulis kode atau instruksi untuk program komputer yang dapat melakukan berbagai macam fungsi, seperti mengolah data, mengontrol perangkat keras, atau menyediakan layanan online. Programmer dapat bekerja di berbagai bidang, seperti pengembangan web, aplikasi seluler, game, perangkat lunak, atau sistem informasi. Programmer juga dapat bekerja secara mandiri sebagai freelancer, atau bergabung dengan tim pengembangan perangkat lunak di perusahaan, lembaga, atau organisasi.

Ada banyak jenis programmer yang berbeda, tergantung pada bahasa pemrograman, teknologi, dan platform yang digunakan. Beberapa jenis programmer yang populer adalah front-end developer, back-end developer, full-stack developer, dan mobile developer. Masing-masing jenis programmer memiliki spesialisasi dan tanggung jawabnya sendiri dalam mengembangkan program komputer. Misalnya, front-end developer bertanggung jawab untuk membuat antarmuka pengguna yang interaktif dan responsif, sedangkan back-end developer bertanggung jawab untuk mengelola logika bisnis dan basis data yang berjalan di sisi server.

Untuk menjadi programmer yang handal, diperlukan berbagai keterampilan teknis dan non-teknis. Keterampilan teknis meliputi kemampuan untuk menguasai bahasa pemrograman, algoritma, struktur data, paradigma pemrograman, dan teknik debugging. Keterampilan non-teknis meliputi kemampuan untuk berpikir logis, kreatif, dan analitis, berkomunikasi dengan baik, bekerja sama dengan tim, dan belajar secara mandiri. Selain itu, programmer juga harus selalu mengikuti perkembangan terbaru di bidang teknologi informasi, serta menerapkan standar kualitas dan keamanan dalam pengembangan perangkat lunak.

Tulisan ini bertujuan untuk memberikan gambaran umum tentang programmer, mulai dari pengertian, jenis-jenis, tugas, tanggung jawab, hingga keterampilan yang dibutuhkan. Tulisan ini juga akan membahas beberapa tantangan dan peluang yang dihadapi oleh programmer di masa depan, serta memberikan beberapa tips dan saran untuk menjadi programmer yang sukses. Ruang lingkup dari tulisan ini terbatas pada programmer yang bergerak di bidang pengembangan web dan aplikasi seluler, dengan menggunakan bahasa pemrograman Java, Python, dan JavaScript.

2. METODE

Dalam membangun sebuah project, programmer dapat menggunakan berbagai metode pengembangan perangkat lunak. Metode-metode tersebut dapat membantu programmer dalam mengelola proyek dan memastikan bahwa proyek tersebut dapat diselesaikan dengan baik dan tepat waktu. Beberapa metode pengembangan perangkat lunak yang umum digunakan antara lain:

1. Waterfall: Metode ini merupakan metode pengembangan perangkat lunak tradisional yang sistematis. Metode ini memiliki lima tahapan proses, di antaranya Communication, Planning, Modeling, Construction, dan Deployment .
2. Agile: Metode ini sangat populer dan cocok digunakan untuk proyek yang bisa dipecah menjadi beberapa bagian. Dalam Agile, interaksi antartim dan penyesuaian atas perubahan menjadi prioritas .
3. RAD: Metode ini merupakan singkatan dari Rapid Application Development. Metode ini memungkinkan pengembangan perangkat lunak yang cepat dan efektif. Metode ini terdiri dari empat tahap, yaitu Requirements Planning, User Design, Construction, dan Cutover .
4. Prototyping: Metode ini memungkinkan programmer untuk membuat prototipe perangkat lunak yang dapat diuji oleh pengguna. Metode ini memungkinkan programmer untuk mengumpulkan umpan balik dari pengguna dan memperbaiki perangkat lunak sebelum produk akhir dirilis .

3. HASIL DAN PEMBAHASAN

HTML dan CSS ada dalam dua bahasa yang sangat penting dalam pengembangan web. HTML adalah singkatan dari Hypertext Markup Language dan digunakan untuk membuat struktur halaman web. Sementara CSS adalah singkatan dari Cascading Style Sheets dan digunakan untuk mengatur tampilan halaman web .

Dalam HTML, elemen-elemen seperti teks, gambar, video, dan link ditandai dengan tag. Tag ini kemudian digunakan oleh browser untuk menampilkan konten pada halaman web. Sementara itu, CSS digunakan untuk mengatur tampilan elemen-elemen tersebut. Dengan CSS, dapat mengubah warna, ukuran, jenis font, dan banyak lagi .

3.1. HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Webpage Design</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="main">
    <div class="navbar">
      <div class="icon">
        <h2 class="logo">Shandy</h2>
      </div>

      <div class="menu">
        <ul>
          <li><a href="#">HOME</a></li>
          <li><a href="#">ABOUT</a></li>
          <li><a href="#">SERVICE</a></li>
          <li><a href="#">DESIGN</a></li>
          <li><a href="#">CONTACT</a></li>
        </ul>
      </div>

      <div class="search">
        <input class="srch" type="search" name="" placeholder="Type
To Text">

        <a href="#"><button class="btn">Search</button></a>
      </div>

    </div>
    <div class="content">
      <h1>Web Design & <br><span>Development</span> <br>Course</h1>
      <p class="par">Lorem ipsum, dolor sit amet consectetur
adipisicing elit. <br> Beatae deserunt laborum iste illum animi deleniti,
incididunt distinctio <br> ex sapiente itaque rerum nisi. Molestias id
accusamus, <br> dolorum sapiente facere totam quam.</p>

      <button class="cn"><a href="#">JOIN US</a></button>

      <div class="form">
        <h2>Login Here</h2>
        <input type="email" name="email" placeholder="Enter
Email Here">

        <input type="password" name="" placeholder="Enter
Password Here">

```

```
        <button class="btnn"><a href="#">Login</a></button>

        <p class="link">Don't have an account<br>
        <a href="#">Sign up </a> here</p>
        <p class="liw">Log in with</p>
    </div>
</div>
</div>
</body>
</html>
```

3.2. CSS

```
* {
    margin: 0;
    padding: 0;
}

.main {
    width: 100%;
    background: linear-gradient(to top, rgba(0,0,0,0.5)50%,
    rgba(0,0,0,0.5)50%), url(dian-yu-CzP6Esb87oQ-unsplash.jpg);
    background-position: center;
    background-size: cover;
    height: 109vh;
}

.navbar {
    width: 1200px;
    height: 75px;
    margin: auto;
}

.icon {
    width: 200px;
    float: left;
    height: 70px;
}

.logo {
    color: orange;
    font-size: 35px;
    font-family: arial;
    padding-left: 20px;
    float: left;
}
```

```
padding-top: 10px;
}

.menu {
width: 400px;
float: left;
height: 70px;
}

ul {
float: left;
display: flex;
justify-content: center;
align-items: center;
}

ul li {
list-style: none;
margin-left: 62px;
margin-top: 27px;
font-size: 14px;
}

ul li a {
text-decoration: none;
color: white;
font-family: arial;
font-weight: bold;
transition: 0.4s ease-in-out;
}

ul li a:hover {
color: orange;
}

.search {
width: 330px;
float: left;
margin-left: 270px;
}

.srch {
font-family: 'Times New Roman';
width: 200px;
height: 40px;
background: transparent;
```

```
border: 1px solid orange;
margin-top: 13px;
color: white;
border-right: none;
font-size: 16px;
float: left;
padding: 10px;
border-bottom-left-radius: 5px;
border-top-left-radius: 5px;
}

.btn {
width: 100px;
height: 40px;
background: orange;
border: 2px solid orange;
margin-top: 13px;
color: white;
font-size: 15px;
border-bottom-right-radius: 5px;
border-bottom-right-radius: 5px;
}

.btn:focus {
outline: none;
}

.srch:focus {
outline: none;
}

.content {
width: 1200px;
height: auto;
margin: auto;
color: white;
position: relative;
}

.content .par {
padding-left: 20px;
padding-bottom: 25px;
font-family: arial;
letter-spacing: 1.2px;
line-height: 30px;
}
```

```
.content h1 {
  font-family: 'Times New Roman';
  font-size: 50px;
  padding-left: 20px;
  margin-top: 9%;
  letter-spacing: 2px;
}

.content .cn {
  width: 160px;
  height: 40px;
  background: orange;
  border: none;
  margin-bottom: 10px;
  margin-left: 20px;
  font-size: 18px;
  border-radius: 10px;
  cursor: pointer;
  transition: 2s ease;
}

.content .cn a {
  text-decoration: none;
  color: black;
  transition: 3s ease;
}

.cn:hover {
  background-color: white;
}

.content span {
  color: orange;
  font-size: 60px;
}

.form {
  width: 250px;
  height: 380px;
  background: linear-gradient(to top, rgba(0,0,0,0.8)50%,
  rgba(0,0,0,0.8)50%);
  position: absolute;
  top: -20px;
  left: 870px;
  border-radius: 10px;
```

```
padding: 25px;
}

.form h2 {
width: 220px;
font-family: sans-serif;
text-align: center;
color: orange;
font-size: 22px;
background-color: white;
border-radius: 10px;
margin: 2px;
padding: 8px;
}

.form input {
width: 240px;
height: 35px;
background: transparent;
border-bottom: 1px solid orange;
border-top: none;
border-right: none;
border-left: none;
color: white;
font-size: 15px;
letter-spacing: 1px;
margin-top: 30px;
font-family: sans-serif;
}

.form input:focus {
outline: none;
}

::placeholder {
color: white;
font-family: arial;
}

.btnn {
width: 240px;
height: 40px;
background: orange;
border: none;
margin-top: 30px;
font-size: 18px;
```



```
border-radius: 10px;
cursor: pointer;
color: white;
transition: 0.4s ease;
}

.btnn:hover {
background-color: white;
color: orange;
}

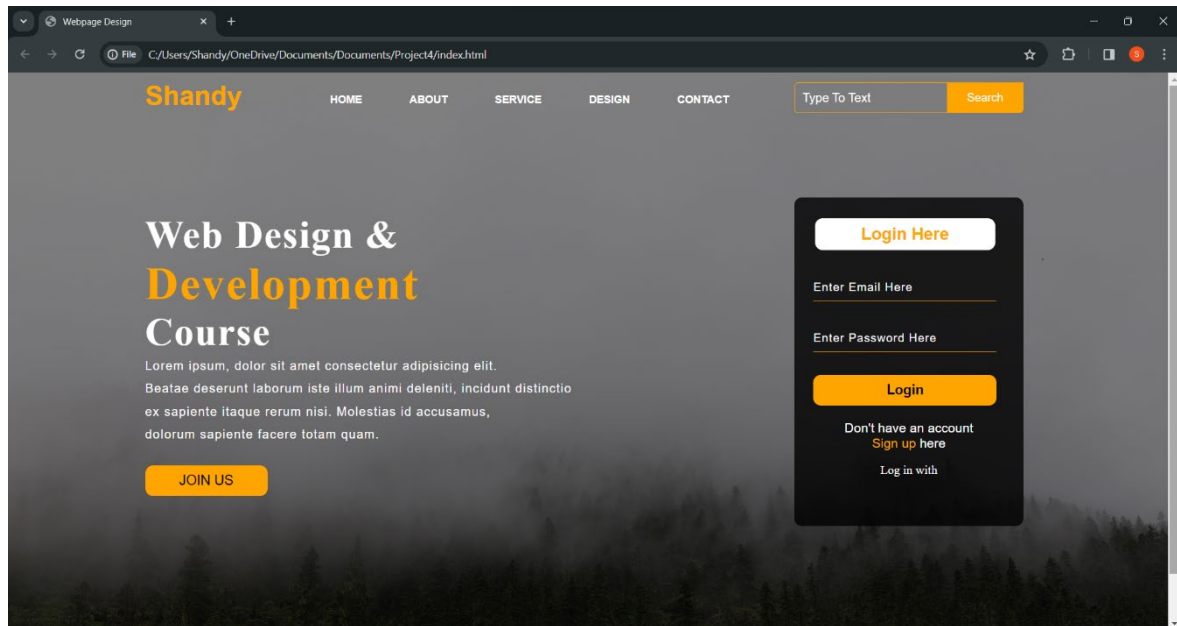
.btnn a {
text-decoration: none;
color: black;
font-weight: bold;
}

.form .link {
font-family: Arial, Helvetica, sans-serif;
font-size: 17px;
padding-top: 20px;
text-align: center;
}

.form .link a {
text-decoration: none;
color: orange;
}

.liw {
padding-top: 15px;
padding-bottom: 10px;
text-align: center;
}
```

3.3. Tampilan Hasil HTML & CSS



4. KESIMPULAN

Dengan menggabungkan HTML dan CSS memungkinkan untuk merancang tampilan web yang menarik dan responsif. Dalam HTML, elemen-elemen seperti teks, gambar, video, dan link ditandai dengan tag. Tag ini kemudian digunakan oleh browser untuk menampilkan konten pada halaman web. Sementara itu, CSS digunakan untuk mengatur tampilan elemen-elemen tersebut. Dengan CSS pengguna dapat mengubah warna, ukuran, jenis font, dan banyak lagi.

Untuk menggabungkan HTML dan CSS, pengguna dapat menambahkan CSS pada file HTML dengan tag `<style>` atau menggunakan file external dengan tag `<link>`. Memahami struktur HTML dan CSS, menggunakan CSS selector, dan memahami CSS inheritance dan cascade adalah hal penting yang harus diperhatikan saat menggabungkan HTML dan CSS.

Kesimpulannya, menggabungkan HTML dan CSS memungkinkan pengguna merancang tampilan web yang menarik dan responsif. Dengan kombinasi keduanya, pengguna dapat menciptakan tampilan yang sesuai dengan visi desain pengguna dan memberikan pengalaman yang baik kepada pengunjung, terlepas dari perangkat yang mereka gunakan.

REFERENCES

- [1] <https://dailysocial.id/post/programmer-adalah>
- [2] <https://sinta.unud.ac.id/uploads/wisuda/1208605081-2-BAB%20I.pdf>
- [3] <https://www.sekawanmedia.co.id/blog/metode-software-development/>
- [4] <https://glints.com/id/lowongan/metode-project-management/>
- [5] <https://medium.com/@IskaWeb/menggabungkan-html-dan-css-menciptakan-tampilan-web-yang-menarik-dan-responsif-c39c5a3d9661>
- [6] <https://pemburukode.com/cara-menggabungkan-html-dan-css/>