

**MATERI PEMBELAJARAN
HTML, CSS, JAVASCRIPT
ELEMEN PEMROGRAMAN WEB**



Oleh

ACHMAD SYAHRUL RAMADHAN

NIM 21050974004

**UNIVERSITAS NEGERI SURABAYA
FAKULTAS TEKNIK
PROGRAM STUDI S1 PENDIDIKAN TEKNOLOGI INFORMASI
2025**

HTML, CSS, JAVASCRIPT

Elemen	: Pemrograman Web
Capaian Pembelajaran	: Pada akhir fase F peserta didik mampu memahami konsep dan menerapkan perintah HTML, CSS, pemrograman Javascript, bahasa pemrograman server-side serta implementasi framework pada pembuatan web statis dan dinamis untuk beragam kebutuhan yang kontekstual. Selain itu, peserta didik juga mampu mendokumentasikan serta mempresentasikan web statis dan dinamis yang telah dikembangkan.
Kelas/Fase	: XI/F
Tujuan Pembelajaran	: <ol style="list-style-type: none">1.1. Peserta didik dapat merancang web interaktif dengan mengintegrasikan HTML, CSS, dan JavaScript berdasarkan studi kasus kehidupan sehari-hari melalui bimbingan LMS, dengan struktur kode yang valid dan menerapkan prinsip berpikir komputasi;1.2. Peserta didik dapat membuat struktur dasar website menggunakan HTML berdasarkan studi kasus di LMS dengan hasil validasi HTML minimal 100% tanpa error;1.3. Peserta didik dapat mengintegrasikan CSS untuk memperindah tampilan website berbasis proyek yang diberikan di LMS dengan kode yang rapi dan keterbacaan minimal 90%;1.4. Peserta didik dapat mengimplementasikan JavaScript dalam proyek website untuk membuat minimal 3 fitur interaktif sesuai panduan LMS dan berfungsi tanpa error

Pendahuluan

Pernahkah kamu berpikir bagaimana aplikasi seperti WhatsApp, Facebook, maupun Instagram bisa mengirim pesan hanya dalam hitungan detik? Atau bagaimana kalkulator di ponselmu bisa langsung memberikan hasil perhitungan dengan cepat? Semua itu dapat terjadi karena adanya pemrograman. Pemrograman adalah proses memberikan instruksi kepada komputer agar dapat menjalankan tugas tertentu. Komputer sebenarnya hanyalah mesin yang tidak bisa berpikir sendiri, ia hanya akan melakukan apa yang diperintahkan melalui baris-baris kode yang ditulis oleh manusia. Dalam dunia teknologi saat ini, pemrograman menjadi keterampilan penting karena hampir semua perangkat digital bekerja berdasarkan program. Dari aplikasi di smartphone, sistem keamanan, hingga mesin-mesin industri semuanya membutuhkan kode program untuk bisa berfungsi dengan baik. Salah satu pendekatan penting dalam pemrograman adalah pemrograman web. Dengan pendekatan ini, pengembang dapat membuat program yang lebih efisien, rapi, dan mudah dikembangkan kembali di masa depan.

Pemrograman Web

A. Definisi Pemrograman

Pemrograman adalah proses menulis, menguji, memperbaiki, dan memelihara kumpulan instruksi (kode program) yang digunakan untuk memberitahu komputer bagaimana cara menyelesaikan suatu tugas tertentu. Instruksi-instruksi ini ditulis dalam bahasa pemrograman seperti C++, Java, Python, Go, dan lain sebagainya. Tujuan dari pemrograman adalah untuk membuat aplikasi, sistem, atau perangkat lunak yang dapat menjalankan perintah tertentu secara otomatis dan efisien. Dalam prosesnya, pemrograman juga

melibatkan pemecahan masalah secara logis dan sistematis agar solusi yang diberikan oleh komputer sesuai dengan kebutuhan pengguna. Pemrograman tidak hanya sekadar menyetik kode, tetapi juga membutuhkan pemahaman tentang logika, struktur, dan alur kerja yang tepat agar program dapat berjalan dengan benar dan optimal.

Setelah memahami apa itu pemrograman, penting bagi siswa untuk mengenal salah satu pendekatan dasar dalam menulis program, yaitu pemrograman web.

B. Pengenalan Aplikasi Berbasis Web dan Pemrograman Web

1. Aplikasi Berbasis Web

Aplikasi atau perangkat lunak (software) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan pengguna (brainware), perangkat keras (hardware) dan jaringan (networking). Jika dilihat dari lingkungan pengembangannya, aplikasi dapat dibagi menjadi aplikasi berbasis desktop, aplikasi berbasis web dan aplikasi berbasis mobile. Pembahasan ini hanya berfokus kepada aplikasi berbasis web. Aplikasi berbasis web adalah aplikasi yang tidak memerlukan instalasi di setiap komputer karena aplikasi berada di suatu server[1]. Untuk membuka aplikasi cukup menggunakan browser yang terhubung melalui jaringan ke server. Situs web merupakan salah satu contoh jenis aplikasi berbasis web. Situs Web, atau yang dikenal sebagai World Wide Web (WWW), adalah sistem yang memungkinkan pengguna untuk mengakses informasi yang saling terhubung melalui jaringan internet. Informasi ini disajikan dalam bentuk dokumen atau halaman web yang menggunakan bahasa markup seperti Hypertext Markup Language (HTML). Web berfungsi sebagai antarmuka yang memudahkan manusia untuk menjelajahi, membaca, dan berinteraksi dengan data yang disediakan[2].

2. Pemrograman Web

Pemrograman web (web programming) terdiri dari kata pemrograman dan web. Pemrograman sendiri dapat diartikan sebagai proses atau cara pembuatan program menggunakan bahasa pemrograman. Adapun bahasa pemrograman merupakan bahasa yang digunakan untuk memberikan intruksi kepada komputer sehingga komputer dapat memproses data dan menampilkan informasi sesuai yang dikehendaki oleh pemrogram. Dengan demikian pemrograman web dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis web menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai yang dikehendaki oleh pemilik website[3].

C. HTML

1. Pengenalan HTML

HTML (HyperText Markup Language) adalah bahasa markah yang digunakan untuk membuat dan menyusun struktur halaman web. HTML adalah fondasi utama dari hampir semua situs web, yang berfungsi untuk menentukan bagaimana elemen-elemen seperti teks, gambar, video, tautan, dan elemen multimedia lainnya ditampilkan pada browser. Berbeda dengan bahasa pemrograman seperti Python atau JavaScript, HTML bukanlah bahasa pemrograman melainkan bahasa markah. Ini berarti HTML berfungsi untuk mendeskripsikan konten, bukan untuk menjalankan logika atau algoritma.

HTML menggunakan elemen-elemen berbentuk tag yang dikelilingi oleh tanda kurung siku (<>) untuk menandai bagian tertentu dari dokumen. Contohnya, tag <p> digunakan untuk membuat paragraf, sedangkan tag <h1> hingga <h6> digunakan untuk membuat judul dengan tingkat kepentingan yang berbeda. Setiap elemen HTML biasanya memiliki tag pembuka (seperti <p>) dan tag penutup (seperti

</p>), meskipun ada beberapa elemen yang tidak memerlukan penutup, seperti untuk gambar. Struktur dasar dokumen HTML diawali dengan deklarasi <!DOCTYPE html>, yang memberitahukan browser bahwa dokumen tersebut menggunakan HTML5.

Salah satu keunggulan HTML adalah kemampuannya untuk mengintegrasikan berbagai teknologi web lain, seperti CSS (Cascading Style Sheets) untuk mempercantik tampilan dan JavaScript untuk menambahkan interaktivitas. Dengan menggabungkan HTML, CSS, dan JavaScript, pengembang dapat menciptakan pengalaman web yang dinamis dan menarik. Selain itu, HTML juga mendukung pengaturan atribut pada elemen-elemen tertentu, seperti memberikan nama kelas (class), mengatur identitas elemen (id), atau menyisipkan tautan dengan atribut href.

Standar HTML terus berkembang seiring waktu melalui pengembangan oleh W3C (World Wide Web Consortium) dan WHATWG (Web Hypertext Application Technology Working Group). Versi terbaru saat ini, HTML5, memperkenalkan fitur-fitur baru seperti elemen semantik (<header>, <footer>, <article>), dukungan untuk multimedia tanpa plugin tambahan (<audio> dan <video>), serta API yang lebih canggih untuk aplikasi web modern. Perkembangan ini dirancang untuk menjawab kebutuhan pengguna web yang semakin kompleks.

Secara keseluruhan, HTML adalah pilar utama dalam pengembangan web. Meskipun tampaknya sederhana, penggunaannya sangat mendalam karena HTML tidak hanya mendefinisikan struktur halaman web tetapi juga berinteraksi dengan teknologi pendukung lainnya. Pemahaman HTML adalah langkah awal yang penting bagi siapa saja yang ingin mempelajari pengembangan web, baik untuk situs pribadi maupun aplikasi web berskala besar.

HTML (HyperText Markup Language) tidak hanya menjadi dasar dalam pembuatan halaman web, tetapi juga memainkan peran penting dalam memastikan aksesibilitas dan kompatibilitas situs web di berbagai perangkat. Dengan HTML, pengembang dapat menyusun halaman web yang responsif, yang dapat disesuaikan tampilannya di layar desktop, tablet, hingga smartphone. Fitur seperti elemen <meta> memungkinkan pengaturan viewport sehingga tata letak halaman dapat beradaptasi sesuai ukuran layar perangkat. Hal ini penting di era modern, di mana penggunaan perangkat seluler untuk mengakses internet semakin mendominasi.

Pada intinya, HTML adalah tulang punggung internet yang memungkinkan kita untuk menikmati informasi dan interaktivitas dalam bentuk situs web yang terstruktur. Tanpa HTML, pengembangan web modern tidak akan mungkin terjadi. Meski terlihat sederhana, memahami cara kerja HTML secara mendalam dan menguasai implementasinya dapat membuka peluang besar bagi pengembang untuk menciptakan pengalaman digital yang luar biasa, mulai dari halaman web statis hingga aplikasi web yang kompleks.

2. Struktur Dasar HTML

Struktur dasar HTML terdiri dari elemen-elemen yang membentuk tulang punggung sebuah dokumen web, dimulai dari deklarasi tipe dokumen hingga isi kontennya. Elemen-elemen tersebut harus mengikuti aturan sintaks tertentu agar halaman dapat di-render dengan benar oleh browser. Berikut ini adalah penjelasan mendetail tentang struktur dasar HTML dalam lima paragraf:

a. Deklarasi Doctype dan Elemen <html>

Struktur HTML selalu diawali dengan deklarasi `<!DOCTYPE html>` yang berfungsi untuk memberi tahu browser bahwa dokumen menggunakan versi HTML5. Deklarasi ini tidak bersifat case-sensitive dan harus ditempatkan di bagian paling atas dokumen. Setelah deklarasi, elemen `<html>` digunakan untuk membungkus seluruh konten dokumen HTML. Elemen ini adalah elemen akar (root element) dari dokumen HTML dan biasanya mencakup atribut `lang` untuk menentukan bahasa utama halaman, misalnya `<html lang="en">` untuk bahasa Inggris atau `<html lang="id">` untuk bahasa Indonesia.

b. Elemen `<head>` untuk Metadata

Bagian pertama dalam elemen `<html>` adalah elemen `<head>`, yang berfungsi untuk menyimpan metadata dan informasi penting lainnya tentang dokumen. Elemen ini tidak ditampilkan langsung di halaman web, tetapi digunakan oleh browser dan mesin pencari. Di dalam `<head>`, terdapat beberapa elemen penting seperti `<title>` (untuk judul halaman yang muncul di tab browser), `<meta>` (untuk informasi seperti encoding karakter dan deskripsi halaman), serta `<link>` (untuk menghubungkan file CSS eksternal) dan `<script>` (untuk menyertakan file JavaScript eksternal atau kode inline).

c. Elemen `<body>` untuk Konten Halaman

Setelah elemen `<head>`, struktur HTML dilanjutkan dengan elemen `<body>`. Elemen ini adalah tempat semua konten yang terlihat oleh pengguna berada. Elemen-elemen seperti heading (`<h1>` hingga `<h6>`), paragraf (`<p>`), gambar (``), tautan (`<a>`), tabel (`<table>`), daftar (`` dan ``), dan elemen lainnya diletakkan di dalam `<body>`. Browser membaca konten dalam `<body>` dan menampilkannya di layar sesuai dengan gaya dan tata letak yang diterapkan menggunakan CSS.

d. Hirarki dan Penutupan Elemen

HTML menggunakan struktur hierarki berbasis elemen yang ditandai dengan tag pembuka dan penutup. Misalnya, elemen `<p>` untuk paragraf ditulis sebagai `<p>Konten paragraf</p>`. Tag pembuka ditulis dengan nama elemen, , sedangkan tag penutup memiliki garis miring sebelum nama elemen. Beberapa elemen, seperti `` atau `
`, adalah elemen kosong (self-closing) yang tidak membutuhkan tag penutup. Setiap elemen dapat bersarang (nested), namun harus sesuai aturan hierarki. Misalnya, elemen `` hanya boleh berisi elemen `` di dalamnya.

e. Validasi dan Aksesibilitas

Struktur HTML yang baik harus divalidasi menggunakan validator seperti W3C Validator untuk memastikan tidak ada kesalahan sintaks. Selain itu, elemen HTML sebaiknya dirancang untuk mendukung aksesibilitas, seperti menggunakan atribut `alt` pada elemen `` untuk memberikan deskripsi gambar bagi pengguna dengan gangguan penglihatan. Struktur yang terorganisasi dengan baik, seperti menggunakan heading dalam urutan logis dan menyediakan landmark dengan atribut `role`, akan membantu mesin pencari dan teknologi bantu memahami konten halaman dengan lebih baik.

f. Elemen Semantik dalam HTML

HTML5 memperkenalkan elemen semantik yang dirancang untuk memberikan makna lebih jelas pada struktur halaman. Contohnya adalah elemen `<header>`, `<footer>`, `<article>`, `<section>`, dan `<aside>`. Elemen-elemen ini membantu browser, mesin pencari, dan pengembang memahami fungsi bagian-bagian dalam dokumen web. Sebagai contoh, `<header>` biasanya digunakan untuk menyimpan judul halaman atau navigasi utama, sementara `<footer>` sering digunakan untuk informasi tambahan seperti hak cipta atau

tautan penting. Elemen semantik tidak hanya mempermudah pembacaan kode, tetapi juga meningkatkan aksesibilitas dan SEO (Search Engine Optimization).

g. Atribut pada Elemen HTML

Setiap elemen HTML dapat memiliki atribut yang memberikan informasi tambahan atau mengontrol perilaku elemen tersebut. Atribut ditulis di dalam tag pembuka, misalnya `Tautan`. Atribut href dalam elemen `<a>` menunjukkan alamat tujuan tautan, sedangkan `target="_blank"` membuat tautan terbuka di tab baru. Selain itu, atribut seperti `id`, `class`, dan `style` sering digunakan untuk identifikasi elemen, pengelompokan elemen untuk styling dengan CSS, atau memberikan gaya langsung pada elemen.

D. CSS

1. Pengenalan CSS

CSS (Cascading Style Sheet) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti font, color, text, dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS, memungkinkan kita untuk menampilkan halaman yang sama dengan format berbeda. Cascading Style Sheet (CSS) juga dapat dinyatakan sebagai sekumpulan aturan yang menyatakan bagaimana style diaplikasikan ke tag-tag HTML di dalam dokumen.

CSS sendiri merupakan sebuah teknologi internet yang direko-mendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996[4]. Untuk saat ini, terdapat tiga versi CSS, yaitu CSS1, CSS2, dan CSS3. CSS1 dikembangkan berpusat pada formatting dokumen HTML, CSS2 dikembangkan untuk bisa memenuhi kebutuhan terhadap format dokumen supaya bisa ditampilkan di printer, CSS3 dikembangkan untuk mendukung desain responsif dan memungkinkan pengembang untuk menciptakan pengalaman pengguna yang lebih dinamis dan interaktif. CSS mendukung seluruh browser, baik Google Chrome, Mozilla Firefox, Microsoft Edge, dan lain sebagainya.

2. Menghubungkan CSS dengan HTML

Cara menghubungkan CSS dengan HTML terdapat tiga metode, diantaranya:

- a. Embedded: properti style diletakkan di dalam satu blok di dokumen HTML. Contoh penerapan sebagai berikut:

```
p {  
  color: red;  
  font-style: italic;  
}
```

- b. Inline: properti style diterapkan secara langsung per baris atau per elemen HTML. Contoh penerapan sebagai berikut:

```
<p style="color:red; font-style:italic">
```

- c. Linked: properti style diletakkan di file berekstensi css dan dikaitkan dengan dokumen HTML. Contoh penerapan sebagai berikut:

```
<link rel="stylesheet" href="style.css" type="text/css" />
```

dimana,

- a) `<link`, merupakan tag pembuka diakhiri dengan tanda `>`;

- b) rel="stylesheet", menerangkan halaman ini akan dikenai efek style sheet;
- c) type="text/css", file yang dipanggil berupa css;
- d) href="style.css", alamat dokumen stylesheet yang dipanggil;

3. Struktur Dasar CSS

Struktur dasar CSS terdiri atas elemen-elemen yang membentuk tulang punggung sebuah style dokumen web. Elemen-elemen tersebut harus mengikuti aturan sintaks tertentu agar halaman dapat di-render dengan benar oleh browser. Berikut ini sintaks pada CSS.

$$\textit{Selector} \{ \textit{Property} : \textit{Value}; \}$$

dimana, *selector* adalah tag HTML yang akan didefinisikan, misalnya h1 dan p. *Property* adalah atribut yang akan diubah. *Value* adalah nilai dari properti yang ingin diubah.

4. Properti CSS Dasar

- 1) Warna: color, background-color
- 2) Font: font-size, font-family, text-align
- 3) Box Model: margin, padding, border
- 4) Layout: display: flex;, grid, position
- 5) Responsif: @media query

5. Selektor CSS

- 1) Elemen: p { color: green; }
- 2) Class: .kotak { border: 1px solid; }
- 3) ID: #judul { font-size: 24px; }
- 4) Pseudo-class: button:hover { background: yellow; }

6. Desain Web Responsif

Desain Web Responsif atau *Responsive Web Design* (RWD) adalah pendekatan desain web untuk membuat halaman web ditampilkan dengan baik di semua ukuran dan resolusi layar sekaligus memastikan kegunaan yang baik. Ini adalah cara mendesain web untuk berbagai perangkat. Sebelum desain web responsif menjadi pendekatan standar untuk membuat situs web berfungsi di berbagai jenis perangkat, pengembang web biasa berbicara tentang desain web seluler, pengembangan web seluler, atau terkadang, desain yang ramah seluler. Ini pada dasarnya sama dengan desain web responsif — tujuannya adalah untuk memastikan bahwa situs web berfungsi dengan baik di berbagai perangkat dengan atribut fisik yang berbeda (ukuran layar, resolusi) dalam hal tata letak, konten (teks dan media), dan kinerja.

Desain web responsif, atau RWD, adalah pendekatan desain yang membahas berbagai perangkat dan ukuran perangkat, yang memungkinkan adaptasi otomatis ke layar, baik konten dilihat di tablet, ponsel, televisi, atau jam tangan. Desain web responsif bukanlah teknologi yang terpisah — melainkan sebuah pendekatan. Ini adalah istilah yang digunakan untuk menggambarkan serangkaian praktik terbaik yang digunakan untuk membuat tata letak yang dapat merespons perangkat apa pun yang digunakan untuk melihat konten. Istilah desain responsif, dicetuskan oleh Ethan Marcotte pada tahun 2010, menggambarkan penggunaan kisi-kisi yang lancar, gambar yang lancar, dan kueri media untuk membuat konten yang responsif[5].

E. Javascript

1. Pengenalan Javascript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “LiveScript” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman “Java”) pada masa itu, maka Netscape memberikan nama “JavaScript” kepada bahasa tersebut pada tanggal 4 Desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “Jscript” di browser Internet Explorer 3.

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengekseskuan perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web.

Javascript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScript selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien.

2. Penulisan Javascript

Kode Javascript dituliskan pada file HTML. Terdapat dua cara untuk menuliskan kode-kode Java Script agar dapat ditampilkan pada halaman HTML, yaitu Javascript ditulis pada file yang sama atau Javascript ditulis pada file terpisah.

a. Javascript ditulis pada file yang sama

Untuk penulisan dengan cara ini, perintah yang digunakan adalah `<script> ... (program javascript disini) </script>`. Perintah tersebut biasanya diletakkan diantara tag `<body> ... </body>`.

b. Javascript ditulis pada file terpisah

Kode Javascript bisa juga kita buat dalam file terpisah dengan tujuan agar dokumen HTML isinya tidak terlalu panjang. Untuk penulisan dengan cara ini, perintah yang digunakan adalah `<script src="namafile.js"> ... (program javascript disini) </script>`.

3. Sintaks Dasar Javascript

a. Variabel

```
let nama = "Andi";  
let umur = 17;  
if (umur >= 17) {  
  alert("Sudah cukup umur");  
}
```

Dalam bahasa pemrograman, variabel digunakan untuk menyimpan nilai data. JavaScript menggunakan kata kunci **var**, **let** dan **const** untuk mendeklarasikan variabel. Tanda sama dengan digunakan untuk menetapkan nilai pada variabel.

b. Fungsi

```
function sapa() {  
  alert("Halo Dunia!");  
}
```

Fungsi JavaScript adalah blok kode yang dirancang untuk melakukan tugas tertentu. Fungsi JavaScript dieksekusi ketika "sesuatu" memanggilnya (memanggilnya).

c. Manipulasi DOM (Document Object Model)

```
<button onclick="ubahTeks()">Klik Saya</button>  
<p id="teks">Teks awal</p>  
  
<script>  
  function ubahTeks() {  
    document.getElementById("teks").innerHTML = "Teks berubah!";  
  }  
</script>
```

DOM adalah antarmuka pemrograman (API) yang memungkinkan program dan skrip (seperti JavaScript) untuk mengakses dan memodifikasi konten, struktur, dan gaya sebuah dokumen HTML atau XML secara dinamis.

d. Event Handling

- 1) onclick adalah event handling untuk memanggil fungsi saat tombol diklik;
- 2) onchange adalah event handling untuk memanggil fungsi saat pengguna mengubah opsi yang dipilih dari elemen <select>;
- 3) onmouseover adalah event handling untuk memanggil fungsi saat menggerakkan penunjuk tetikus (mouse) ke gambar;
- 4) addEventListener adalah event handling untuk penanganan peristiwa ke suatu elemen;

F. Prosedur Pengembangan Aplikasi Berbasis Web

Software Development Life Cycle (SDLC) atau dalam bahasa Indonesia Siklus Hidup Pengembangan Sistem merupakan salah satu metode pengembangan perangkat lunak, terutama aplikasi berbasis web yang bertujuan untuk menyelesaikan masalah secara efektif. Dalam pengertian lain, SDLC adalah tahapan kerja yang bertujuan untuk menghasilkan sistem berkualitas tinggi yang sesuai dengan

keinginan pelanggan atau tujuan dibuatnya sistem tersebut [6]. Menurut Dwanoko, 2018, SDLC adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi [7]. SDLC menjadi kerangka yang berisi langkah-langkah yang harus dilakukan untuk memproses pengembangan suatu perangkat lunak. Sistem ini berisi rencana lengkap untuk mengembangkan, memelihara, dan menggantikan perangkat lunak tertentu.

SDLC digunakan untuk membangun suatu sistem informasi agar dapat berjalan sesuai dengan apa yang diharapkan. SDLC (Systems Development Life Cycle, Siklus Hidup Pengembangan Sistem) atau Systems Life Cycle (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana(planning), analisis (analysis), desain (design), implementasi (implementation), uji coba (testing) dan pengelolaan (maintenance).

1. Perencanaan Sistem (Systems Planning)

Lebih menekankan pada aspek studi kelayakan pengembangan sistem (feasibility study). Aktivitas-aktivitas yang ada meliputi:

- a. Pembentukan dan konsolidasi tim pengembang.
- b. Mendefinisikan tujuan, ruang lingkup pengembangan, dan target pengguna.
- c. Mengidentifikasi apakah masalah-masalah yang ada bisa diselesaikan melalui pengembangan sistem.
- d. Menentukan dan evaluasi strategi yang akan digunakan dalam pengembangan sistem.
- e. Penentuan prioritas teknologi dan pemilihan aplikasi.

2. Analisis Sistem (Systems Analysis)

Pada tahap ini, sistem akan dianalisis bagaimana akan dijalankan nantinya. Hasil analisis berupa kelebihan dan kekurangan sistem, fungsi sistem, hingga pembaharuan yang dapat diterapkan. Bagian ini termasuk dalam bagian perencanaan. Bagian lain yang termasuk dalam perencanaan ialah alokasi sumber daya, perencanaan kapasitas, penjadwalan proyek, estimasi biaya, dan penetapan. Dengan demikian, hasil dari tahap perencanaan ialah rencana proyek, jadwal, estimasi biaya, dan ketentuan. Idealnya manajer proyek dan pengembang dapat bekerja maksimal pada tahap ini.

Analisa sistem adalah tahap di mana dilakukan beberapa aktivitas berikut:

- a. Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani oleh sistem.
- b. Brainstorming dalam tim pengembang mengenai kasus mana yang paling tepat dimodelkan dengan sistem.
- c. Mengklasifikasikan masalah, peluang, dan solusi yang mungkin diterapkan untuk kasus tersebut.
- d. Analisa kebutuhan pada sistem dan membuat batasan-batasan sistem.
- e. Mendefinisikan kebutuhan fungsional dan non-fungsional sistem.

3. Perancangan Sistem (Systems Design)

Tahapan ini akan menghasilkan prototype dan beberapa output lain meliputi dokumen berisi desain, pola, dan komponen yang diperlukan untuk mewujudkan proyek tersebut. Setelah spesifikasi, kemudian dilakukan perancangan sistem sebagai tahapan kelanjutannya. Tahap ini ialah tahap di mana seluruh hasil analisis dan pembahasan tentang spesifikasi sistem diterapkan menjadi rancangan atau cetak biru sebuah sistem. Tahap ini disebut sebagai cetak biru, di mana sistem sudah siap untuk dikembangkan mulai dari implementasi, analisis sistem, hingga tenaga pendukung sistem yang akan dikembangkan.

Pada tahap ini, features dan operasi-operasi pada sistem dideskripsikan secara detail. Aktivitas-aktivitas yang dilakukan adalah:

- a. Menganalisa interaksi obyek dan fungsi pada sistem.
- b. Menganalisa data dan membuat skema database.
- c. Merancang user interface dan/atau user experiences.

4. Implementasi Sistem (Systems Implementation)

Tahap berikutnya adalah implementasi yaitu mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Dalam implementasi, dilakukan aktivitas-aktivitas sebagai berikut:

- a. Pembuatan database sesuai skema rancangan.
- b. Pembuatan aplikasi berdasarkan desain sistem.
- c. Pengujian dan perbaikan aplikasi (debugging).

5. Pemeliharaan Sistem (Systems Maintenance)

Dilakukan oleh admin yang ditunjuk untuk menjaga sistem tetap mampu beroperasi secara benar melalui kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan.

G. Project Charter

1. Pengenalan Project Charter

Project Charter di dalam manajemen proyek adalah dokumen resmi yang menginisiasi sebuah proyek dengan memberikan otorisasi kepada manajer proyek untuk memulai dan mengerjakan proyek tersebut [8]. Dokumen ini memuat informasi penting seperti tujuan proyek, scope, anggota tim, serta wewenang manajer proyek. Dalam konteks manajemen proyek, Project Charter berfungsi sebagai panduan dan referensi sepanjang siklus hidup proyek.

2. Elemen-Elemen Dalam Project Charter

- a. Judul Proyek: Menyediakan identifikasi proyek yang unik dan memudahkan referensi.
- b. Latar Belakang dan Tujuan Proyek: Menjelaskan alasan mengapa proyek ini diperlukan dan tujuan utama yang ingin dicapai.
- c. Ruang Lingkup Proyek (Project Scope): Mendefinisikan batasan proyek, termasuk apa yang termasuk dan tidak termasuk dalam proyek.
- d. Anggaran dan Sumber Daya: Menguraikan anggaran yang dialokasikan untuk proyek dan sumber daya yang dibutuhkan.

- e. Risiko dan Asumsi: Mengidentifikasi potensi risiko yang mungkin dihadapi dan asumsi-asumsi yang mendasari perencanaan proyek.
- f. Tim Proyek: Menyebutkan anggota tim proyek dan peran serta tanggung jawab mereka.
- g. Jadwal Proyek: Menyediakan jadwal keseluruhan proyek, termasuk milestone dan tenggat waktu penting.

3. Langkah-Langkah Membuat Project Charter

- a. Kumpulkan Informasi Dasar: Mulailah dengan mengumpulkan informasi dasar mengenai proyek, seperti tujuan, latar belakang, dan ruang lingkup.
- b. Identifikasi Pemangku Kepentingan: Identifikasi siapa saja yang terlibat dalam proyek dan apa peran serta tanggung jawab mereka.
- c. Definisikan Tujuan dan Sasaran: Tentukan tujuan dan sasaran proyek yang spesifik, terukur, dapat dicapai, relevan, dan berbatas waktu (SMART).
- d. Buat Jadwal dan Anggaran: Susun jadwal proyek yang mencakup milestone penting dan buat anggaran yang realistis berdasarkan sumber daya yang tersedia.
- e. Identifikasi Risiko dan Asumsi: Lakukan analisis risiko untuk mengidentifikasi potensi hambatan dan buat daftar asumsi yang digunakan dalam perencanaan proyek.
- f. Review dan Validasi: Review Project Charter dengan pemangku kepentingan utama dan pastikan semua informasi sudah akurat dan lengkap.
- g. Dapatkan Persetujuan: Dapatkan persetujuan dari sponsor proyek atau manajemen senior untuk mengesahkan Project Charter.

H. Praktikum Berdasarkan Studi Kasus

STUDI KASUS: Sekolah kamu ingin meningkatkan minat siswa terhadap kegiatan ekstrakurikuler melalui media digital. Saat ini, belum ada sistem digital untuk mendaftar kegiatan ekstrakurikuler, sehingga prosesnya masih manual dan menyulitkan siswa maupun guru pembina. Sebagai seorang calon pengembang web profesional, kamu diminta menganalisis kebutuhan pengguna, lalu merancang dan membangun sebuah website interaktif yang digunakan untuk pendaftaran ekstrakurikuler secara online.

PROSEDUR PENYELESAIAN (Menggunakan Project-Based Learning & Berpikir Komputasi):

(Orientasi Pada Masalah)

1. **Membuat Rumusan Masalah (*Decomposition*):** Bagaimana cara merancang dan membangun sebuah website interaktif untuk pendaftaran ekstrakurikuler sekolah secara online, yang mudah digunakan dan tidak bergantung pada database eksternal?
2. **Membuat Indikator Penyebab (*Decomposition*):** Proses pendaftaran ekstrakurikuler masih dilakukan secara manual, kurangnya media informasi digital tentang ekstrakurikuler sekolah, siswa kesulitan mengakses informasi dan formulir secara efisien.
3. **Membuat Indikator Penyelesaian (*Pattern Recognition*):** Website menampilkan informasi ekstrakurikuler dengan menarik, website menyediakan formulir online yang dapat diisi oleh siswa, Data input ditampilkan langsung (tanpa database), menggunakan JavaScript.

(Perencanaan dan Penjadwalan Proyek)

4. Menentukan Spesifikasi Proyek, Pembagian Tugas Kelompok Jika Berkelompok, dan Menyusun Langkah dan Jadwal Pembuatan Website (*Abstraction dan Algorithm*):

- Bahasa pemrograman atau teknologi yang digunakan yaitu HTML, CSS, Javascript.
- Fitur utama website yaitu informasi ekstrakurikuler, form pendaftaran, tampilan hasil input.
- Target pengguna website yaitu siswa SMK.
- Media/alat yang digunakan yaitu Visual Studio Code dan Web Browser seperti Google Chrome.
- Batas waktu pengerjaan website yaitu sepekan (misalnya 15 Mei s.d. 22 Mei 2025).

(Pembuatan Proyek)

5. Membuat Kode Website (*Abstraction dan Algorithm*):

Contoh Kode (index.html):

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Pendaftaran Ekstrakurikuler</title>
</head>
<body>

  <!-- Header website -->
  <header>
    <h1>Pendaftaran Ekstrakurikuler Sekolah</h1>
  </header>

  <!-- Konten utama -->
  <main>
    <h2>Formulir Pendaftaran</h2>

    <!-- Formulir pendaftaran -->
    <form id="formDaftar">
      <!-- Input nama -->
      <label for="nama">Nama Lengkap</label>
      <input type="text" id="nama" name="nama" required>

      <!-- Input kelas -->
      <label for="kelas">Kelas</label>
      <input type="text" id="kelas" name="kelas" placeholder="Contoh:
XI RPL 3" required>

      <!-- Pilihan ekstrakurikuler -->
      <label for="ekstra">Pilih Ekstrakurikuler</label>
      <select id="ekstra" name="ekstra" required>
        <option value="">-- Pilih Ekstrakurikuler --</option>
        <option value="Basket">Basket</option>
        <option value="Pramuka">Pramuka</option>
        <option value="IT Club">IT Club</option>
        <option value="Paskibra">Paskibra</option>
      </select>

      <!-- Tombol kirim -->
      <button type="submit">Daftar Sekarang</button>
    </form>
  </main>
</body>
</html>
```

```

</form>

<!-- Hasil pendaftaran ditampilkan di sini -->
<div class="result" id="hasilPendaftaran"></div>
</main>

</body>
</html>

```

Contoh Kode (style.css):

```

* {
  box-sizing: border-box;
  font-family: Arial, sans-serif;
}

/* warna background halaman */
body {
  margin: 0;
  padding: 0;
  background-color: #f0f4f8;
}

/* Header website */
header {
  background-color: #0066cc;
  color: white;
  padding: 20px;
  text-align: center;
}

/* Kontainer utama */
main {
  max-width: 600px;
  margin: 40px auto;
  background: white;
  padding: 30px;
  border-radius: 12px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

/* Judul subbagian */
h2 {
  margin-top: 0;
}

/* Label form */
label {
  display: block;
  margin-top: 15px;
  font-weight: bold;
}

/* Input dan select form */
input[type="text"],
select {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
}

```

```

border: 1px solid #ccc;
border-radius: 6px;
}

/* Tombol submit */
button {
margin-top: 20px;
width: 100%;
padding: 12px;
background-color: #28a745;
color: white;
font-size: 16px;
border: none;
border-radius: 6px;
cursor: pointer;
}

/* Efek hover pada tombol */
button:hover {
background-color: #218838;
}

/* Kotak hasil pendaftaran */
.result {
margin-top: 20px;
padding: 10px;
background-color: #e6f7ff;
border: 1px solid #91d5ff;
border-radius: 6px;
display: none; /* awalnya disembunyikan */
}

/* Responsif untuk layar kecil */
@media (max-width: 600px) {
main {
margin: 20px;
padding: 20px;
}
}

```

Contoh Kode (script.js):

```

document.getElementById("formDaftar").addEventListener("submit",
function(event) {
event.preventDefault(); // Mencegah reload halaman

// Ambil nilai input dari form
const nama = document.getElementById("nama").value.trim();
const kelas = document.getElementById("kelas").value.trim();
const ekstra = document.getElementById("ekstra").value;

// validasi sederhana jika ada yang kosong
if (!nama || !kelas || !ekstra) {
alert("Semua kolom wajib diisi!");
return;
}
}

```

```

// Tampilkan hasil pendaftaran ke pengguna
const hasil = document.getElementById("hasilPendaftaran");
hasil.style.display = "block";
hasil.innerHTML =
    <strong>Terima kasih, ${nama}!</strong><br>
    Kamu telah berhasil mendaftar pada ekstrakurikuler
<strong>${ekstra}</strong>.<br>
    Kami akan menghubungi kamu lebih lanjut melalui wali kelas
    ${kelas}.
;

// Kosongkan kembali form
document.getElementById("formDaftar").reset();
});

```

Hasil tampilan *user interface* (UI):

The image shows a web form titled "Pendaftaran Ekstrakurikuler Sekolah". The form is titled "Formulir Pendaftaran" and contains three input fields: "Nama Lengkap" (a text box), "Kelas" (a text box with the example "Contoh: XI RPL 3"), and "Pilih Ekstrakurikuler" (a dropdown menu with "-- Pilih Ekstrakurikuler --"). Below the fields is a green button labeled "Daftar Sekarang".

Gambar 1.1. Tampilan UI Praktikum Web

- 6. Pengujian dan Debugging:** Pengujian dalam hal ini berkaitan dengan fungsional sebuah website yang meliputi pengujian input form, validasi data, memperbaiki error, mengidentifikasi kesalahan logika dan tampilan, lalu memperbaikinya secara iteratif, dan lain sebagainya.

I. Latihan Soal

STUDI KASUS: Saat ini penggunaan AI seperti ChatGPT, Gemini, Copilot, dan lainnya meningkat drastis di kalangan pelajar, terutama untuk menyelesaikan tugas secara instan. Di media sosial, ramai perdebatan tentang bagaimana AI dimanfaatkan oleh siswa—antara membantu pembelajaran atau justru membuat ketergantungan. Banyak guru dan sekolah mulai menyuarakan pentingnya literasi digital dan etika penggunaan AI. Kasus viral seperti “jawaban tugas disalin dari ChatGPT tanpa dipahami” menjadi sorotan.

DESKRIPSI PROYEK: Siswa diminta membuat sebuah website edukatif yang bertujuan memberikan pemahaman kepada teman sebaya tentang AI generatif, manfaat dan risiko penggunaannya di dunia pendidikan, tips menggunakan AI secara bijak, peraturan atau etika penggunaan AI di sekolah. Buatlah penyelesaian studi kasus sesuai dengan langkah-langkah yang telah dijelaskan di atas!

DURASI PROYEK: Satu bulan

Daftar Pustaka

- [1] A. Solichin, *Pemrograman Web dengan PHP dan MySQL*. Penerbit Budi Luhur, 2016. [Online]. Available: https://books.google.co.id/books?id=k8-GDAAAQBAJ&dq=Pemrograman+web&lr=&hl=id&source=gbs_navlinks_s
- [2] B. Sisephaputra *et al.*, *Buku Ajar Pemrograman Web*. PT. Sonpedia Publishing Indonesia, 2025. [Online]. Available: https://www.google.co.id/books/edition/Buku_Ajar_Pemrograman_Web/Bnw9EQAAQBAJ?hl=id&gbpv=1&dq=Buku+Ajar+Pemrograman+Web&pg=PA1&printsec=frontcover
- [3] R. Abdulloh, *7 in 1 Pemrograman Web untuk Pemula*. Jakarta: PT Elex Media Komputindo, 2018. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=21FwDwAAQBAJ&oi=fnd&pg=PP1&dq=Pemrograman+web&ots=ppJ4zkzrIE&sig=UyBCak5qBXrDYQ0kuCFNU7JAFnw&redir_esc=y#v=onepage&q=Pemrograman+web&f=false
- [4] D. Juju and Matamaya, *Kupas Tuntas CSS*. Elex Media Komputindo, 2013. [Online]. Available: https://www.google.co.id/books/edition/Kupas_Tuntas_CSS/7-BMDwAAQBAJ?hl=id&gbpv=1
- [5] M. Contributors, “Responsive Design,” *Mozilla Corporation*, 2025. https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design (accessed Apr. 28, 2025).
- [6] B. Nusantara, “Memahami System Development Life Cycle,” *Binus University*, 2020. <https://accounting.binus.ac.id/2020/05/19/memahami-system-development-life-cycle/> (accessed Apr. 28, 2025).
- [7] Y. S. Dwanoko, “Implementasi Software Development Life Cycle (SDLC) Dalam Penerapan Pembangunan Aplikasi Perangkat Lunak,” *J. Teknol. Inf.*, vol. 7, no. 2, 2016, [Online]. Available: <https://www.neliti.com/publications/143003/implementasi-software-development-life-cycle-sdlc-dalam-penerapan-pembangunan-ap>
- [8] IBIKKG, “Pengertian Project Charter dan Pentingnya dalam Manajemen Proyek,” *kwikkiangie*, 2024. <https://kwikkiangie.ac.id/2024/07/26/pengertian-project-charter-dan-pentingnya-dalam-manajemen-proyek/#:~:text=Apa+itu+Project+Charter%3F,memulai+dan+mengerjakan+proyek+tersebut.> (accessed Apr. 28, 2025).