

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Survei yang dilakukan di Universitas Telkom menunjukkan bahwa 79,3% dari 58 responden hanya menyelesaikan kurang dari lima proyek latihan dalam jangka waktu 3-6 bulan. Temuan ini menandakan kurangnya penerapan aktif dan praktis dari pengetahuan pemrograman yang didapatkan. Fenomena ini sering disebut sebagai *Tutorial Hell* [1], yaitu kondisi di mana pemrogram pemula merasa telah menguasai pemrograman hanya dengan menonton atau mengikuti tutorial, tanpa benar-benar mempraktikkan atau menerapkan apa yang telah dipelajari dalam proyek nyata. Hal ini menciptakan ilusi kemahiran, padahal keterampilan nyata dalam pemrograman memerlukan lebih dari sekadar pemahaman teoritis [2].

Mengatasi masalah ini memerlukan pendekatan belajar pemrograman yang lebih interaktif dan praktis [3], maka diperlukan solusi berupa aplikasi pembelajaran pemrograman dengan fitur latihan untuk memfasilitasi pembelajaran interaktif dan praktis. Platform yang digunakan adalah web karena menyediakan aksesibilitas yang luas, memungkinkan pengguna untuk mengakses aplikasi dari berbagai perangkat dan lokasi, menghilangkan batasan geografis dan memperluas jangkauan pembelajaran pemrograman. Selain itu, interaksi yang efisien melalui *keyboard* pada platform web memberikan kemudahan bagi pengguna dalam menulis, mengedit, dan menguji kode, aspek penting yang mendukung proses pembelajaran yang efektif. Terakhir, ketersediaan perangkat lunak pemrograman dan alat pengembangan yang dapat diakses tanpa perlu instalasi rumit sangat membantu bagi pemrogram pemula, yang mungkin belum memiliki pengetahuan atau kemampuan untuk mengatur lingkungan pengembangan yang kompleks [4]. Dengan demikian, platform web terpilih sebagai platform ideal untuk aplikasi yang akan dibuat, menyediakan lingkungan pembelajaran yang inklusif, mudah diakses, dan mengedepankan pengalaman belajar pemrograman yang lebih interaktif dan praktis.

1.2 Rumusan Masalah

Bagaimana cara memfasilitasi programmer pemula untuk menerapkan pengetahuan pemrograman yang baru mereka pelajari?

1.3 Tujuan

Adapun tujuan dari proyek akhir ini adalah untuk mengembangkan aplikasi belajar pemrograman yang memiliki fitur eksekusi dan pengetesan otomatis. Sehingga membantu programmer pemula menerapkan dan menguji pengetahuan pemrograman yang baru dipelajari secara langsung dalam lingkungan yang Interaktif.

1.4 Batasan Masalah

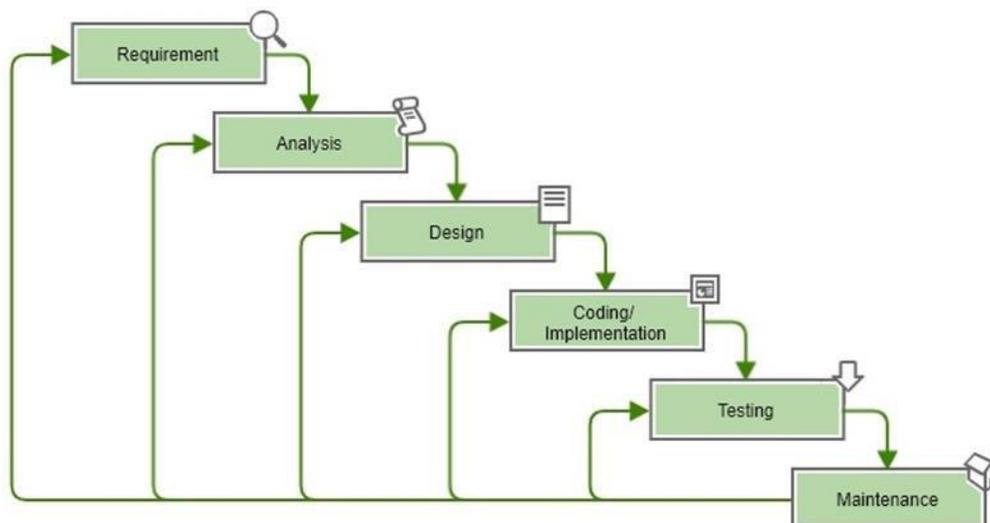
Batasan masalah dalam proyek akhir ini, yaitu:

1. Platform usulan hanya menggunakan platform *website*.
2. Aplikasi yang dibuat akan mengajarkan bahasa pemrograman Javascript.
3. Konten pembelajaran hanya berbentuk artikel.

1.5 Metode Pengerjaan

Untuk membangun aplikasi ini, menerapkan dua metode pengembangan: SDLC Waterfall lalu Inkremental. Metode Waterfall digunakan pada tahap awal pengembangan yang akan menghasilkan versi pertama dari aplikasi. Setelah itu, proyek beralih ke metode Inkremental untuk memfasilitasi perkembangan berkelanjutan dan penyesuaian terhadap kebutuhan pengguna.

1.5.1 Metode Pengembangan Waterfall

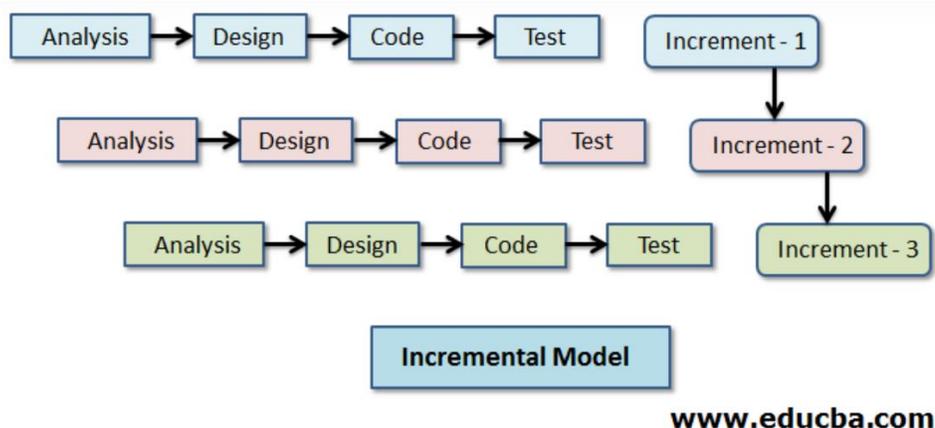


Gambar 1 Tahapan Pada SDLC Waterfall [13]

Dalam proses pengembangan perangkat lunak menggunakan metode SDLC Waterfall, akan dilalui beberapa tahap sebagai berikut:

1. **Requirement:** Pada tahap ini akan didefinisikan masalah yang dihadapi dan merancang solusi berdasarkan identifikasi masalah. Pada tahap ini dilakukan survei dengan metode kuesioner yang bertujuan untuk mengkonfirmasi masalah *tutorial hell*. Survei dilakukan ke mayoritas mahasiswa Universitas Telkom dan beberapa Karyawan. Pada tahap ini juga dilakukan perincian fitur yang akan dikembangkan
2. **Design:** Pada tahap ini, dibuat desain dari solusi yang akan dikembangkan. Desain dibuat menggunakan Figma.
3. **Implementation:** Pada tahap ini dilakukan pengembangan aplikasi, di mana setiap fitur-fitur yang sudah didefinisikan pada tahap sebelumnya mulai diimplementasikan secara bertahap, dengan memastikan bahwa setiap fitur berfungsi sesuai dengan desain dan spesifikasi awal.
4. **Verification:** Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem. Pada tahap ini akan dilakukan *usability testing*. Di mana pengguna akan dihadapkan pada tugas-tugas dalam aplikasi dan menilai apakah aplikasi mudah digunakan.
5. **Maintenance:** Tahap *maintenance* pada pengembangan waterfall digantikan dengan metode inkremental karena metode inkremental memungkinkan pengembangan produk secara bertahap, dengan fitur-fitur baru ditambahkan dan disempurnakan secara berurutan. Ini memfasilitasi perbaikan berkelanjutan dan penyesuaian terhadap perubahan kebutuhan atau masukan pengguna, yang lebih efisien dibandingkan pendekatan sekali selesai pada model waterfall.

1.5.2 Metode Pengembangan Inkremental



Gambar 2 Tahapan Pada SDLC Inkremental [6]

Dalam proses pengembangan perangkat lunak menggunakan metode inkremental akan dilakukan hal berikut pada setiap langkahnya:

1. **Analysis:** Pada tahap ini dilakukan analisis terhadap hasil *usability testing* yang sudah dilakukan pada langkah verifikasi dalam tahap pengembangan menggunakan metode *waterfall*. Jika yang dilakukan sebelumnya adalah tahap inkremental maka yang di analisis adalah hasil dari serangkaian pengujian yang dilakukan di tahap *Test*. Hasil dari analisis kemudian akan menjadi pertimbangan pada tahap berikutnya
2. **Design:** Dalam tahap ini, akan ditentukan dan dirancang solusi untuk kekurangan yang telah diidentifikasi selama tahap analisis. Tahap ini melibatkan pembuatan rencana pembuatan atau perbaikan fitur secara rinci.
3. **Code:** Setelah desain selesai, tahap berikutnya adalah implementasi desain tersebut dalam kode program. Pada tahap ini, pengembang menulis kode untuk mewujudkan fitur dan fungsi yang telah direncanakan.
4. **Test:** Setelah desain selesai diimplementasi, pengujian *usability testing* dilakukan untuk menguji kelayakan dari fitur yang dikembangkan atau ditingkatkan. Pada tahap pengujian ini responden akan diminta untuk menggunakan fitur latihan dengan mengikuti instruksi yang akan diberikan saat pelaksanaan pengujian. Data yang akan didapatkan dari *usability testing* adalah
 - a. Tingkat kemudahan menggunakan fitur latihan: Setelah responden mengerjakan tugas untuk menggunakan fitur latihan, responden akan diminta untuk menjawab apakah fitur latihan tersebut mudah, biasa saja, atau sulit untuk digunakan.
 - b. Saran dan kritik: Setelah melakukan *usability testing*, Responden diperkenankan memberikan kritik dan saran. Hal ini bertujuan untuk mengumpulkan umpan balik langsung dari pengguna tentang pengalaman mereka menggunakan fitur latihan. Data yang didapat digunakan untuk menyimpan saran fitur atau laporan kesalahan program serta memastikan bahwa fitur latihan yang dikembangkan sesuai dengan kebutuhan dan ekspektasi pengguna.
 - c. NPS (*Net Promoter Score*): NPS merupakan metode untuk mengukur seberapa besar kecenderungan pengguna untuk merekomendasikan aplikasi kepada orang lain. Pertanyaan NPS akan muncul secara berkala setelah pengguna menggunakan aplikasi selama durasi tertentu. Pengguna akan diminta untuk menilai dalam skala 0-10 seberapa besar kemungkinan mereka merekomendasikan aplikasi kepada teman atau rekan. Skor ini membantu mengidentifikasi pengguna yang sangat puas (*Promoters*), puas namun tidak sepenuhnya (*Passives*), dan tidak puas (*Detractors*).

○

1.6 Jadwal Pengerjaan

Tabel 1 Jadwal Pengerjaan Proyek

Tahap	Okt 23	Nov 23	Des 23	Jan 24	Feb 24	Mar 24	Apr 24	Mei 24	Jun 24
<i>Waterfall: Requirement</i>									
<i>Waterfall: Implementation</i>									
<i>Waterfall: Verification</i>									
Inkremental 1									
Inkremental 2									
Pembuatan Konten Pembelajaran									
Penyusunan Proyek Akhir									