

BAB II

LANDASAN TEORI

A. User Interface (UI)

Antarmuka pengguna adalah bagian dari sistem informasi yang membutuhkan interaksi pengguna untuk membuat input dan output. Dari dua definisi di atas dapat disimpulkan bahwa antarmuka pengguna adalah bagian yang membentuk dialog antara pengguna dan program, sehingga masukan yang diberikan oleh klien diubah menjadi keluaran yang sesuai dari program itu sendiri. [4].

B. Metode *Prototype*

Prototype atau Purnarupa merupakan simulasi dari semua aspek produk yang akan dikembangkan. Purnarupa adalah metode pengembangan memungkinkan perancang dan pengguna untuk mengevaluasi sistem, sehingga perancang dan pengguna dapat dengan mudah menemukan kelebihan dan kekurangannya. Metode Purnarupa memiliki beberapa tahapan, antara lain [5]:

1. Pengumpulan Kebutuhan

Ditahap ini pengembang mengumpulkan kebutuhan dari sistem yang akan dibuat, biasanya setelah pelanggan atau customer melakukan wawancara langsung yang nantinya menjadi pengguna sistem yang dibuat.

2. Membangun Purnarupa

Pada fase ini, pengembang membuat sebuah Purnarupa atau model dari kebutuhan yang telah dijelaskan oleh pelanggan menggunakan bahasa pemrograman tertentu dan algoritma yang sesuai.

3. Evaluasi Purnarupa

Pada langkah ini, pengembang mempresentasikan Purnarupa tersebut kepada pelanggan, jika Purnarupa yang di evaluasi tidak memenuhi keinginan pelanggan, maka pelanggan dapat memberikan komentar atau modifikasi kepada pengembang apa saja yang harus ditambahkan atau

dihapus dari sistem yang akan dibuat dan pengembang akan membuat ulang Purnarupa dengan hasil dari revisi sebelumnya.

4. Mengkodekan Sistem

Pada fase ini, Purnarupa yang melewati tahap evaluasi dan disetujui oleh pelanggan, kemudian dirakit menjadi sistem yang lengkap dengan bahasa pemrograman dan algoritma yang sesuai kebutuhan.

5. Menguji Sistem

Sistem yang dibuat kemudian memasuki tahap uji, pelanggan dan pengembang melihat seberapa baik sistem yang dibuat, dapat memberikan hasil yang memenuhi persyaratan atau tidak, apakah kinerjanya bagus atau tidak dan banyak hal untuk penggunaan jangka panjang.

6. Evaluasi Sistem

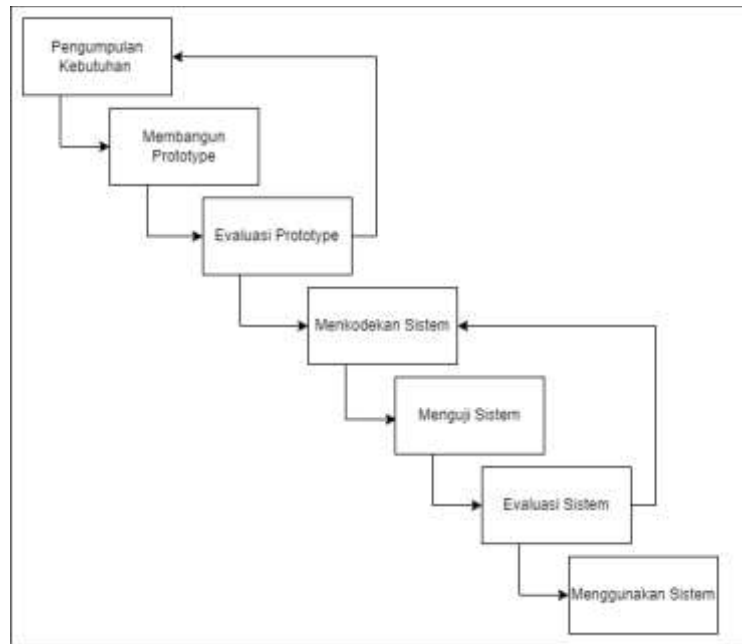
Setelah sistem dievaluasi, kemudian sistem dievaluasi oleh pelanggan maupun pengembang. Jika sistem sebelumnya yang diuji kurang, maka pengembang akan algoritma dibangun kembali sesuai evaluasi. Jika dianggap memadai, maka sistem akan siap tersedia bagi pelanggan.

7. Menggunakan Sistem

Setelah sistem berhasil lulus evaluasi, langkah selanjutnya adalah menggunakan sistem. Sistem digunakan sesuai dengan target awal dibuatnya sistem tersebut .

Pada pembahasan ini hanya akan melakukan tahap 1 dan 2 karena hanya merancang desain *low fidelity* dan belum sampai kepada pembahasan evaluasi desain Purnarupa.

Berikut merupakan alur atau tahapan metode Purnarupa:



Gambar 2. 1 Tahapan Metode Purnarupa [6]

C. Wireframe

Wireframe adalah sketsa awal atau sketsa kasar untuk perakitan atau pengorganisasian elemen pada halaman aplikasi, sebuah proses yang dilakukan sebelum proses desain yang sebenarnya dimulai. Elemen yang dapat disesuaikan misalnya *banner*, *header*, *content*, *footer*, *link*, *dari feed* dan lain-lain. *Wireframe* ini digunakan untuk memfasilitasi, mengarahkan, dan memfasilitasi revisi atau perbaikan. Bayangkan membuat aplikasi dengan mendesainnya langsung tanpa desain atau wireframe asli. Kesalahan manufaktur sering terjadi, menambah waktu atau mempersulit pekerjaan yang dimaksud. Ada kekurangan atau perbaikan yang diinginkan oleh pelanggan, hal-hal tersebut sudah dapat menjadi alasan bagi pembuat aplikasi untuk menyelesaikan langkah-langkah besi sehingga desain aplikasi tidak harus bekerja dua kali. [7].

Banyak yang berpendapat *wireframe*, Purnarupa dan *mockup* adalah suatu hal yang sama, *wireframe* adalah termasuk golongan *low fidelity*, *low fidelity* adalah desain yang tingkat persepsinya masih rendah dan tujuannya hanya untuk menunjukkan tata letak. Sedangkan Purnarupa dan *mockup* termasuk golongan *high fidelity* dimana desain yang tingkat persesinya tinggi, sudah memiliki warna, ukuran, jarak dan bentuk elemen lainnya dengan kata

lain lebih mendetail [8].

1. High-fidelity Design

High fidelity membuat Purnarupa final yang terlihat seperti produk rilis final. Pada tahap ini desain yang sudah sampai pada tahap akhir dapat diberikan kepada tester atau pengguna untuk mendapatkan jawaban apakah model yang didapat cocok, jika pengguna datang dengan versi, maka proses ini kembali ke level rendah. pengembalian akurasi. agar hasil yang memuaskan dapat tercapai. Dalam produksi figma yang tinggi biasanya kita menggunakan aplikasi figma karena dengan bantuan aplikasi figma kita dapat membuat tampilan dan fungsi sesuai dengan layar meskipun tidak ada aliran data di layar. [9].

2. Low-fidelity Design

Low Fidelity membuat Purnarupa yang masih dalam tahap draft dan masih mentah. Tugas prototipe Lo-Fi adalah memeriksa dan menguji fungsionalitas, bukan optik produk. Keuntungan dari *low fidelity* adalah murah karena prototipe masih dalam konsep, lalu cepat karena memungkinkan Anda melakukan *low fidelity* dalam 10 menit, dan kemudian kolaboratif karena Anda melakukannya tidak diperlukan pengetahuan khusus serendah ini. . kesetiaan Dalam penelitian ini saya menggunakan desain low fidelity [9].

D. Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metoda metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP) [10]. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain [10]:

1. *Usecase* Diagram

Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna [11].

2. *Activity* Diagram

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsifungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek [11].

Pada penelitian ini kita hanya akan melakukan desain *Use case diagram*, sedangkan desain selanjutnya dilakukan pada *role* selanjutnya.