BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian berisi uraian yang menjelaskan secara rinci tentang bagaimana suatu penelitian dilakukan. Penjelasan ini juga bisa digunakan untuk menilai apakah hasil suatu penelitian dapat dipercaya, dan apakah kesimpulan penelitian ini dapat digunakan untuk kepentingan secara praktis atau digunakan dalam kajian pustaka dalam penelitian berikutnya

3.2 Desain Penelitian

Desain penelitian merupakan rencana untuk memilih sumber-sumber dan jenis informasi yang dipakai untuk menjawab pertanyaan penelitian.

Desain penelitian menurut Moh. Nazir (2003:84) memaparkan bahwa "desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian".

Dari definisi di atas maka dapat dikatakan bahwa desai penelitian merupakan semua proses penelitian yang dilakukan oleh penulis dalam melaksanakan penelitian mulai dari perencanaan sampai dengan pelaksanaan penelitian yang dilakukan pada waktu tertentu.

3.3 Jenis dan Metode Pengumpulan Data

Metodologi Penelitian yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

3.3.1 Sumber Data Primer

Data primer merupakan informasi yang diperoleh pertama kali oleh peneliti menyangkut variable yang menjadi tujuan utama penelitian. Untuk mendapatkan data primer metode pengumpulan data yang dipakai adalah kuesioner dan Studi pustaka.

Berikut ini penjelasan metode pengumpulan data yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

a. Studi Pustaka

Pengumpulan data seperti buku-buku referensi yang berhubungan dengan android dan pewaktuan sholat. Selain itu juga dengan cara mengumpulkan literatur, *browsing internet*, jurnal-jurnal dan makalah-makalah yang ada kaitannya dengan topik yang diambil untuk membantu dalam penyajian informasi yang akan di tampilkan.

b. Kuesioner

Teknik pengumpulan data yang di rancang terlebih dahulu dengan cara menggunakan daftar pertanyaan yang akan diberikan kepada responden.

3.3.2 Sumber Data Sekunder

Data sekunder yaitu data yang dikumpulkan oleh orang atau institusi selain peneliti yang melakukan kajian pada saat ini. Data sekunder bisa didapat dari internal maupun eksternal organisasi baik melalui internet maupun publikasi.

3.4 Metode Pendekatan dan Pengembangan Sistem

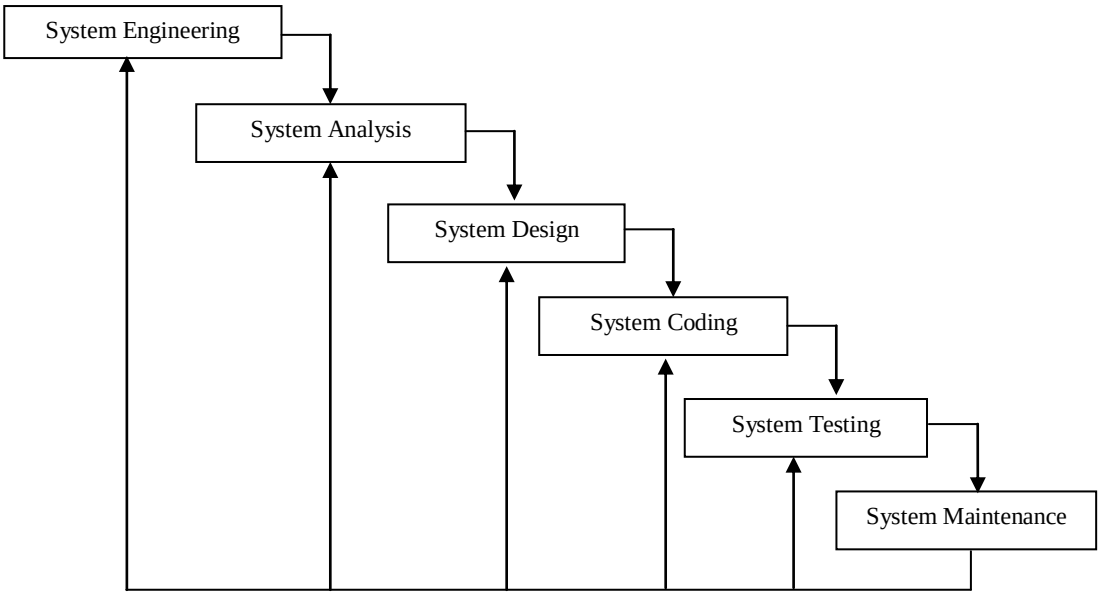
Berikut ini merupakan metode pendekatan dan pengembangan sistem yang dipakai dalam merancang dan membangun perangkat lunak AndroPrayer.

3.4.1 Metode Pendekatan Sistem

Dalam melakukan penelitian ini metode pendekatan yang dipakai adalah pemodelan berorientasi objek karena metode ini dikembangkan sebagai suatu alat analisis dan desain yang sering dipakai dalam pengembangan sistem berbasis *mobile.*

3.4.2 Metode Pengembangan Sitem

Metode pengembangan sistem yang penulis pakai adalah metode *waterfall.* seperti yang tercantum pada gambar 3.1, yang meliputi beberapa proses yaitu sebagai berikut :



Gambar 3.1 Metode *Waterfall*

a. Rekayasa sistem

   Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya kedalam pembentukan perangkat lunak

b. Analisis

   Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak

c. Perancangan

   Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti oleh *user*

d. Pemograman

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman yang telah di tentukan.

e. Pengujian

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun

f. Pemeliharaan

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan–perubahan atau penambahan sesuai dengan permintaan *user*

3.4.3 Alat Bantu Analisis dan Perancangan

Alat bantu Analisis dan perancangan yang dipakai dalam membangun perangkat lunak ini adalah dengan menggunakan pemodelan UML yang merupakan suatu alat untuk analisis dan desain berorientasi objek.

UML menurut Yasmi Afrizal, Wahyuni memaparkan bahwa UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau di hasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML menyediakan 9 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Class Diagram* (Statis)

   Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi. Diagram ini umum ditemui pada pemodelan sistem berorientasi objek. Meski sifatnya statis, sering pula memuat kelas-kelas aktif.

2. *Object Diagrams* (Statis)

   Diagram ini memperlihatkan objek-objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instantiasi statis dari segala sesuatu yang dijumpai dari pada diagram kelas.

3. *Use Case Diagram*

   Diagram ini memperlihatkan himpunan *Use-Case* dan Actor-Actor (jenis khusus dari kelas). Diagram ini penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. *Sequence Diagram* (Dinamis)

   Diagram sequence (urutan) adalah diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

5. *Collaboration Diagrams* (Dinamis)

   Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan (*message*).

6. *State Chart Diagram* (Dinamis)

   Diagram ini memperlihatkan state-state pada sistem; memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sistem reaktif.

7. *Activity Diagram* (Dinamis)

   Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagrams* (Statis)

   Diagram ini memperlihatkan organisasi serta kebergantunngan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.

9. *Deployment Diagrams* (Statis)

   Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*), memuat simpul-simpul atau node beserta komponen-komponen yang ada didalamnya.

3.5 Pengujian

Pengujian merupakan hal penting karena bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang diuji, dan untuk mengetahui apakah perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

3.5.1 Pengujian Alpha

Pengujian alpha dilakukan pada sisi pengembang oleh seorang pelanggan, sebelum perangkat lunak benar-benar jadi. Tujuannya untuk identifikasi dan menghilangkan sebanyak mungkin masalah sebelum perangkat lunak sampai ke pemakai yang sebenarnya.

3.5.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan oleh pemakai akhir dalam lingkungan yang sebenarnya. Tujuannya untuk mendapatkan umpan balik dari pelanggan yang selanjutnya akan dipakai dalam revisi perangkat lunak yang telah dibuat.