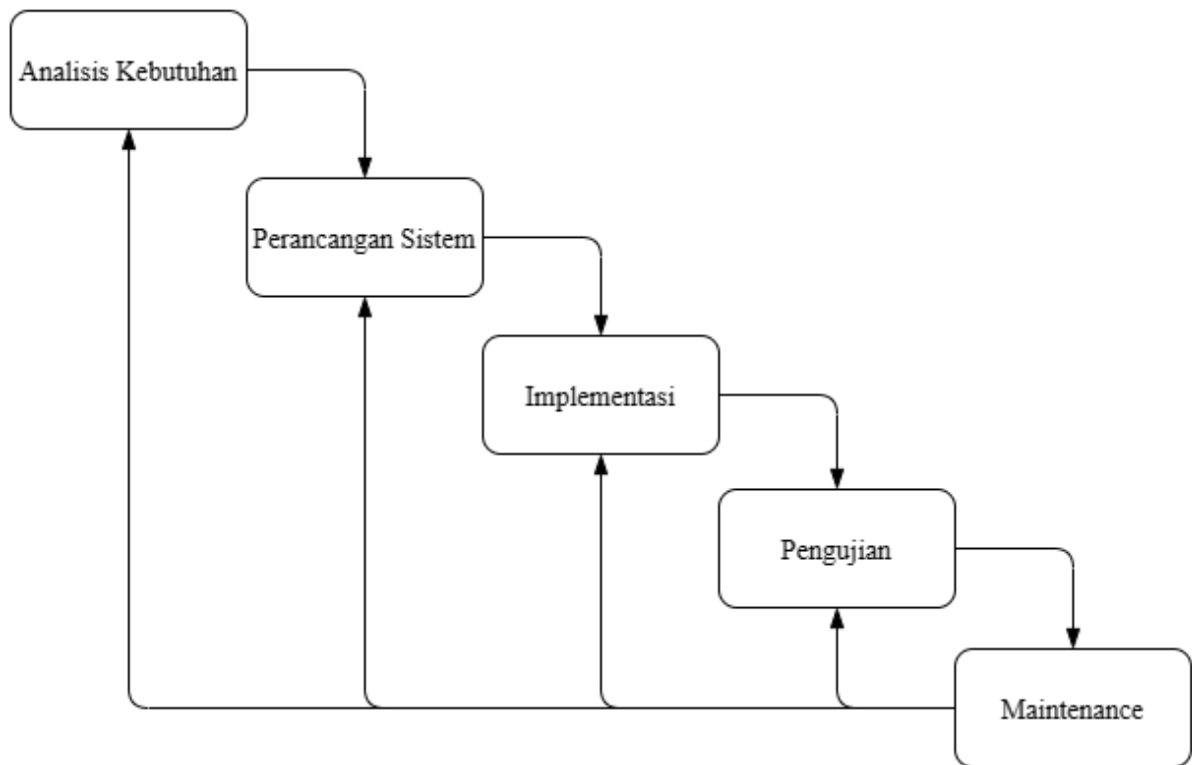


BAB III

METODOLOGI PENELITIAN

Dalam pengembangan Sistem Monitoring Kegiatan Keagamaan di MI Nurrohmah Bina Insani ini, metode waterfall dipilih karena memiliki struktur yang lebih terorganisir dan mempermudah peneliti karena tahapan-tahapan dalam metodenya dapat diidentifikasi dengan jelas dari awal hingga akhir proses pengembangan sistem. Berikut tahapan metode waterfall yang ada pada Gambar 3.1.



Gambar 3.1 *Waterfall* (Ian Sommerville, 2011)

3.1 Analisis Kebutuhan

Analisis kebutuhan sistem dapat dilakukan dengan cara wawancara dan observasi buku monitoring. Ruang lingkup analisis kebutuhan pada Sistem monitoring kegiatan keagamaan MI Nurrohmah Bina Insani sebagai berikut:

a. Kebutuhan Pengguna

Kebutuhan pengguna digunakan sebagai menggambarkan fungsi yang diperlukan oleh pengguna. Pada pengembangan sistem ini kebutuhan pengguna dilakukan untuk mengetahui apa saja pengguna dengan fungsi perannya masing-masing. Dengan adanya pengguna yang memiliki *role* masing-masing dapat mempermudah dalam membuat fitur-fitur yang diperlukan oleh masing-masing pengguna.

b. Kebutuhan Input

Proses untuk memasukan data-data untuk menjadikan data tersebut menjadi sebuah informasi dalam sistem tersebut. Data-data yang dimasukan akan berisikan informasi yang dapat mempermudah pengguna dalam sistem tersebut. Dengan memasukan data-data yang diperlukan data tersebut dapat dijadikan sebuah proses dalam sistem tersebut.

c. Kebutuhan Proses

Tahapan yang dilakukan untuk melanjutkan tahap kebutuhan input. Kebutuhan proses ini akan mengolah data-data yang dimasukan menjadi sebuah informasi. Pengelola ini dilakukan untuk mendapatkan sebuah keluar berupa hasil-hasil dari data yang telah dimasukan.

d. Kebutuhan Keluaran





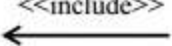
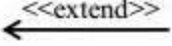
Proses untuk menampilkan informasi yang telah diolah. Informasi yang telah diolah akan berisikan kebutuhan-kebutuhan yang diperlukan oleh pengguna sesuai dengan fitur-fitur yang telah dibuat. Dengan adanya proses keluaran yang berisikan informasi ini akan membantu pengguna dalam memantau proses perkembangan anak.

3.2 Perancangan Sistem

Proses perancangan sistem monitoring kegiatan keagamaan di MI Nurrohmah Bina Insani dilakukan sebagai berikut:

a. Perancangan *Use Case Diagram*

Use case diagram merupakan diagram untuk menjelaskan hubungan interaksi antara aktor atau pengguna sistem dengan sistem itu sendiri. Fungsionalitas sistem digambarkan dalam bentuk diagram sehingga *use case diagram* sering digunakan dalam pengembangan sebuah sistem informasi, yaitu untuk mengidentifikasi ada aktor siapa sedang berinteraksi dengan sistem dan aktor tersebut sedang melakukan aktivitas apa di dalam interaksi dengan sistem. Berikut contoh simbol *Use Case Diagram* pada Gambar 3.2.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

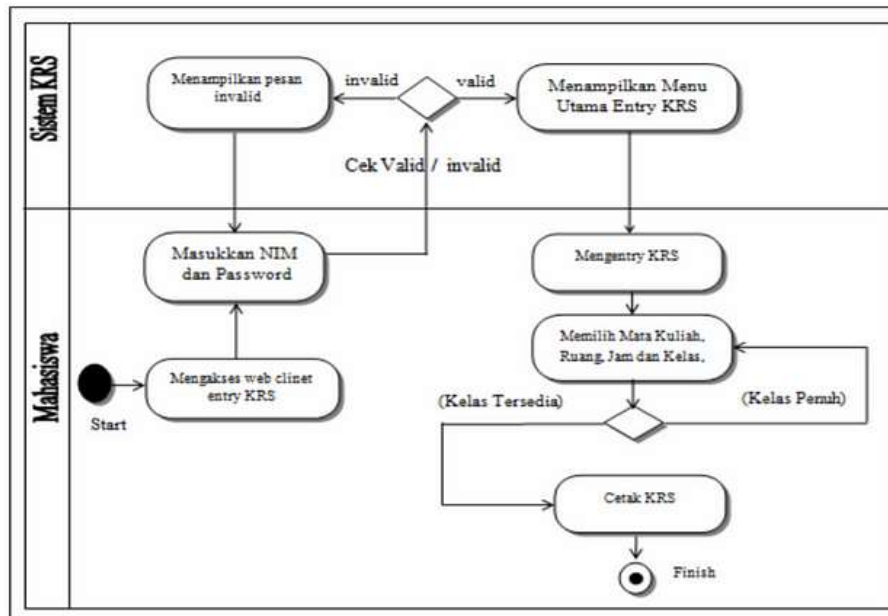
Gambar 3.2 Simbol *Use Case Diagram* (Anita Lasmaria Siagian, 2020)

b. Perancangan *Activity Diagram*

Activity diagram adalah jenis diagram dalam bahasa pemodelan UML yang digunakan untuk merepresentasikan alur kerja atau aktivitas yang dilakukan dalam sistem atau proses bisnis. Diagram ini mencakup aktivitas, tindakan, keputusan, dan kontrol alur.

Activity diagram memungkinkan pengembang perangkat lunak untuk memahami urutan aktivitas dalam suatu proses bisnis atau sistem, dan mengidentifikasi masalah atau proses yang berpotensi mengalami masalah. Diagram ini juga membantu dalam membangun pemahaman tentang peran dan interaksi antara objek-objek yang terlibat dalam proses atau sistem, serta mengoptimalkan alur kerja atau proses bisnis yang ada.

Contoh UML *activity diagram* berikut ini mendeskripsikan proses pendaftaran mahasiswa baru di sebuah universitas pada Gambar 3.3.



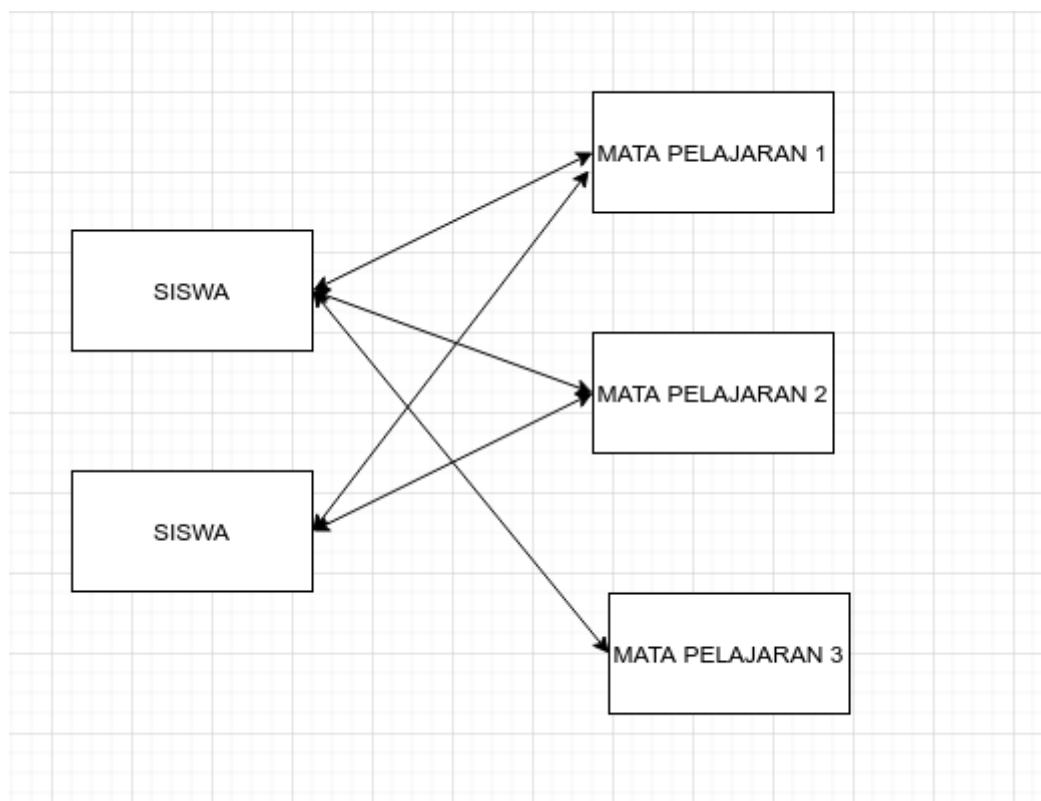
Gambar 3.3 Contoh *Activity Diagram* (Anita Lasmaria Siagian, 2020)

Keterangan :

Activity diagram Mahasiswa diatas menggambarkan interaksi atau kegiatan Mahasiswa antara sistem KRS, sebelum mengentry KRS mahasiswa wajib untuk membayar biaya semesteran, jika mahasiswa belum membayar atau belum melunasi maka tidak mendapat hak akses mahasiswa untuk memasukkan KRS, sebaliknya jika mahasiswa sudah melunasi pembayaran maka akan diberikan hak akses mahasiswa untuk login client di web KRS. Kemudian Mahasiswa memasukkan NIM dan *password*. Kemudian sistem akan mengecek apakah NIM dan *password* yang dimasukkan benar, jika salah, maka sistem akan menampilkan halaman login client lagi dan anggota memasukkan kembali NIM dan *passwordnya*. Apabila NIM dan *password* yang dimasukkan benar, maka akan menampilkan menu utama entry KRS. Data-data tentang perkuliahan sudah terinput ke sistem KRS oleh operator BAAK sehingga mahasiswa tinggal memasukkan KRS sesuai dengan nilai SKS yang telah didapatkannya, lalu memilih mata kuliah, memilih ruangan, jam, hari dan kelas, jika mata kuliah yang dipilih kelasnya penuh mahasiswa harus kembali memilih matakuliah tersebut dengan kelas lain, jika tersedia akan tertampung dikotak bawah lalu jika sudah pilih semua sampai batas sks yang didapat, baru mahasiswa mencetak KRS.

c. Basis Data

Perancangan basis data atau *database* merupakan tempat penyimpanan seluruh data informasi yang digunakan pada sebuah sistem (Rafidah, 2019). Semua informasi yang muncul dalam sistem akan diambil dari basis data. Pada tahap pemasukan dan pencarian data, seluruh proses akan dilakukan di dalam basis data. Basis data yang diterapkan dalam Sistem Monitoring Kegiatan Keagamaan di MI Nurrohmah Bina Insani akan menyimpan semua jenis data, termasuk data siswa, guru, dan orang tua. Perancangan basis data ini dimaksudkan untuk mengoptimalkan pengolahan data yang diperlukan dan memudahkan penampilannya dalam sistem. Perancangan sistem akan dibuat dengan relasi antar tabel. Berikut contoh relasi antar tabel relasi *Many To Many* pada Gambar 3.4.



Gambar 3.4 Contoh relasi antar tabel

d. Perancangan Desain Antarmuka

Menurut Damayanti (2021) perancangan antarmuka merupakan pembuatan rancangan sistem dengan tujuan pengujian dan proses kerja sistem dengan memberikan visualisasi berupa tampilan sistem yang berhubungan dengan masukan, keluaran dan mengolah. Antarmuka juga merupakan representasi visual awal dari suatu sistem yang digunakan sebagai demonstrasi sesuai dengan hasil analisis kebutuhan. Desain antarmuka ini akan disusun menggunakan metode *wireframing*. Perancangan antarmuka ditekankan pada aspek yang mudah digunakan oleh

pengguna, sehingga dapat digunakan dengan sebaik-baiknya. Desain antarmuka ini dibuat sederhana namun mencakup semua fitur yang telah direncanakan.

3.3 Implementasi

Implementasi adalah tahap di mana sistem perangkat lunak yang dapat digunakan oleh pengguna akan dibuat. Dalam konteks Sistem Monitoring Kegiatan Keagamaan di MI Nurrohmah Bina Insani, implementasi akan dilakukan dalam bentuk aplikasi web. Sistem ini akan dibangun sesuai dengan analisis kebutuhan dan perancangan yang telah disusun sebelumnya. Pembuatan sistem ini akan menggunakan bahasa pemrograman PHP dan HTML dengan bantuan *framework Laravel* dan *Bootstrap* untuk mengatur tampilan aplikasi sehingga responsif. Fokus implementasi akan mencakup pengembangan aspek *front end* dan *back end* dari sistem.

3.4 Pengujian

Pengujian sistem ini menggunakan *System Usability Scale (SUS)*. Pengujian ini membutuhkan partisipasi dari kedua belah pihak yaitu pihak pengembang dan pihak pelaksana dari MI Nurrohmah Bina Insani.

System Usability Scale adalah alat pengukuran yang dapat digunakan untuk mengukur tingkat *usability* sebuah sistem. Dibentuk oleh John Brooke pada tahun 1986, *system usability scale* dapat digunakan untuk mengukur tingkat *usability* pada berbagai produk seperti *hardware*, *software*, *mobile app*, hingga *website*.

Beberapa keunggulan menggunakan *system usability scale* antara lain:

- Mudah digunakan dan diterima oleh responden
- Dapat digunakan pada *sample* penelitian yang kecil dengan hasil yang akurat
- Terbukti valid dalam menentukan apakah sistem sudah dapat digunakan dengan baik

System Usability Scale menggunakan skala Likert satu hingga lima yaitu 1 sangat tidak setuju, 2 tidak setuju, 3 netral, 4 setuju, dan 5 sangat setuju. Pertanyaan kuesioner *System Usability Scale* pun perlu disusun secara berurutan yaitu:

1. *I think that I would like to use this system frequently.*

(Saya berpikir akan menggunakan sistem ini lagi.)

2. *I found the system unnecessarily complex.*

(Saya merasa sistem ini rumit untuk digunakan.)

3. *I thought the system was easy to use.*

(Saya merasa sistem ini mudah digunakan.)

4. *I think that I would need the support of a technical person to be able to use this system.*

(Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.)

5. *I found the various functions in this system were well integrated.*

(Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.)

6. *I thought there was too much inconsistency in this system.*

(Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).)

7. *I would imagine that most people would learn to use this system very quickly.*

(Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.)

8. *I found the system very cumbersome to use.*

(Saya merasa sistem ini membingungkan.)

9. *I felt very confident using the system.*

(Saya merasa tidak ada hambatan dalam menggunakan sistem ini.)

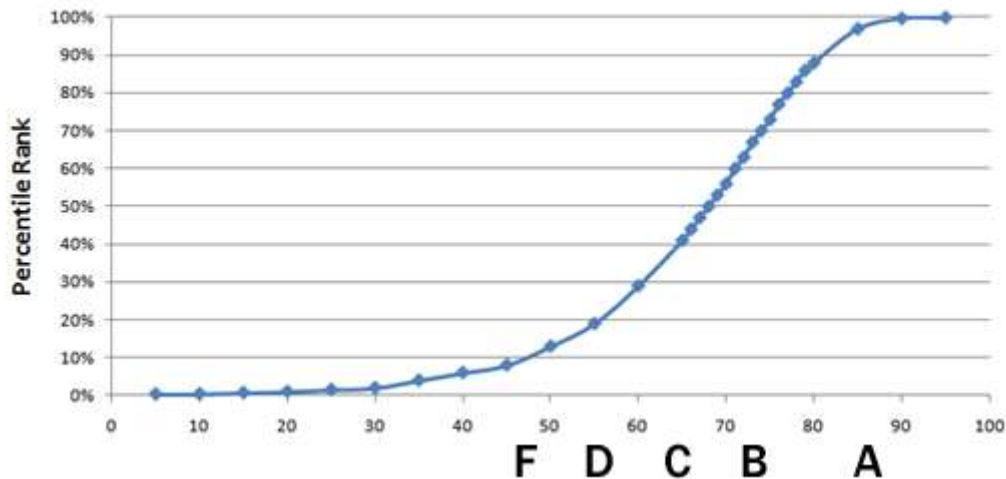
10. *I needed to learn a lot of things before I could get going with this system.*

(Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.)

Adapun cara menghitung hasil pengukuran *system usability scale* yaitu:

- Untuk setiap pertanyaan pada urutan ganjil kurangi dengan nilai satu. Contoh pertanyaan 1 memiliki skor 4. Maka kurangi 4 dengan 1 sehingga skor pertanyaan 1 adalah 3.
- Untuk setiap pertanyaan pada urutan genap kurangi nilainya dari lima. Contoh pertanyaan 2 memiliki skor 1. Maka kurangi 5 dengan 1 sehingga skor pertanyaan 2 adalah 4.
- Tambahkan nilai-nilai dari pernyataan bernomor genap dan ganjil. Kemudian hasil penjumlahan tersebut dikalikan dengan 2,5.

Walau tidak dapat membantu dalam menentukan faktor atau fitur yang masih bermasalah pada sistem, *system usability scale* dapat membantu dalam menentukan apakah sistem sudah dapat digunakan dengan baik (*usability*). Rata-rata tingkat *system usability scale* adalah 68. Maka jika skor dibawah 68 berindikasi terdapat permasalahan yang berpengaruh ke tingkat *usability* sistem. Untuk mengetahui kualitas produk yang dirancang, dirujuk ke grafik precentile rank terhadap SUS score pada Gambar 3.5.



Gambar 3.5 *Percentil Rank* (Susilo, 2019)

Setelah melakukan perhitungan sesuai dengan aturan, kemudian dilakukan perhitungan skor rata-rata dengan rumus berikut (Susilo, 2019).

$$\bar{x} = \frac{\sum x}{n}$$

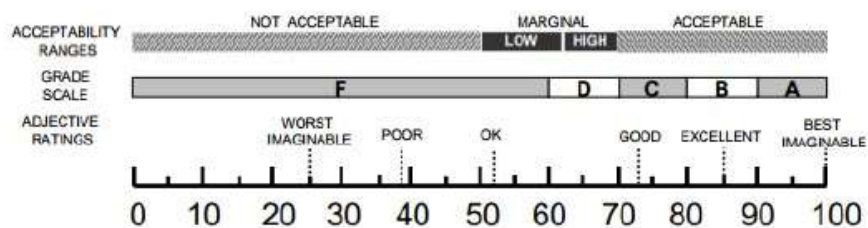
Keterangan:

\bar{x} = skor rata-rata

$\sum x$ = jumlah skor sus

n = jumlah responden

Dalam menentukan kesimpulan dari rata-rata skor sus, dapat dilakukan penyesuaian dengan penilaian SUS sebagai berikut pada Gambar 3.6.



Gambar 3.6 Skor SUS (Susilo, 2019)

3.5 Maintenance

Maintenance *website* adalah kegiatan pemeliharaan *website* yang bertujuan untuk merawat website agar tetap berada pada performa yang baik, terbaru, dan terhindar dari berbagai permasalahan yang dapat merusak atau merugikan.

Tahapan maintenance pada penelitian, khususnya dalam konteks perbaikan error yang ditemukan pada tahap sebelumnya, dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Identifikasi Error

Lakukan identifikasi terhadap error atau kesalahan yang ditemukan pada tahap sebelumnya. Pastikan error tersebut jelas dan dapat direplikasi.

2. Analisis Penyebab

Analisis penyebab terjadinya error tersebut. Apakah disebabkan oleh kesalahan pada metodologi, perangkat lunak, atau faktor lainnya.

3. Perbaikan Error

Setelah penyebab error teridentifikasi, lakukan perbaikan secara sistematis. Misalnya, jika error disebabkan oleh kesalahan pada kode program, lakukan debug dan perbaikan pada kode tersebut.

4. Pengujian

Setelah perbaikan dilakukan, lakukan pengujian untuk memastikan bahwa error telah diperbaiki dengan baik dan tidak muncul kembali. Pengujian ini dapat dilakukan dengan menggunakan data yang sama atau data yang serupa dengan tahap sebelumnya.

5. Validasi

Lakukan validasi terhadap hasil perbaikan untuk memastikan bahwa solusi yang diterapkan benar-benar efektif dalam mengatasi error yang ditemukan.

6. Dokumentasi

Penting untuk mendokumentasikan semua perubahan yang dilakukan pada tahapan maintenance, termasuk identifikasi error, analisis penyebab, perbaikan yang dilakukan, serta hasil pengujian dan validasi.

7. Implementasi

Setelah semua tahapan maintenance selesai dilakukan dan error telah berhasil diperbaiki, implementasikan hasil perbaikan tersebut pada tahapan selanjutnya dari penelitian Anda.