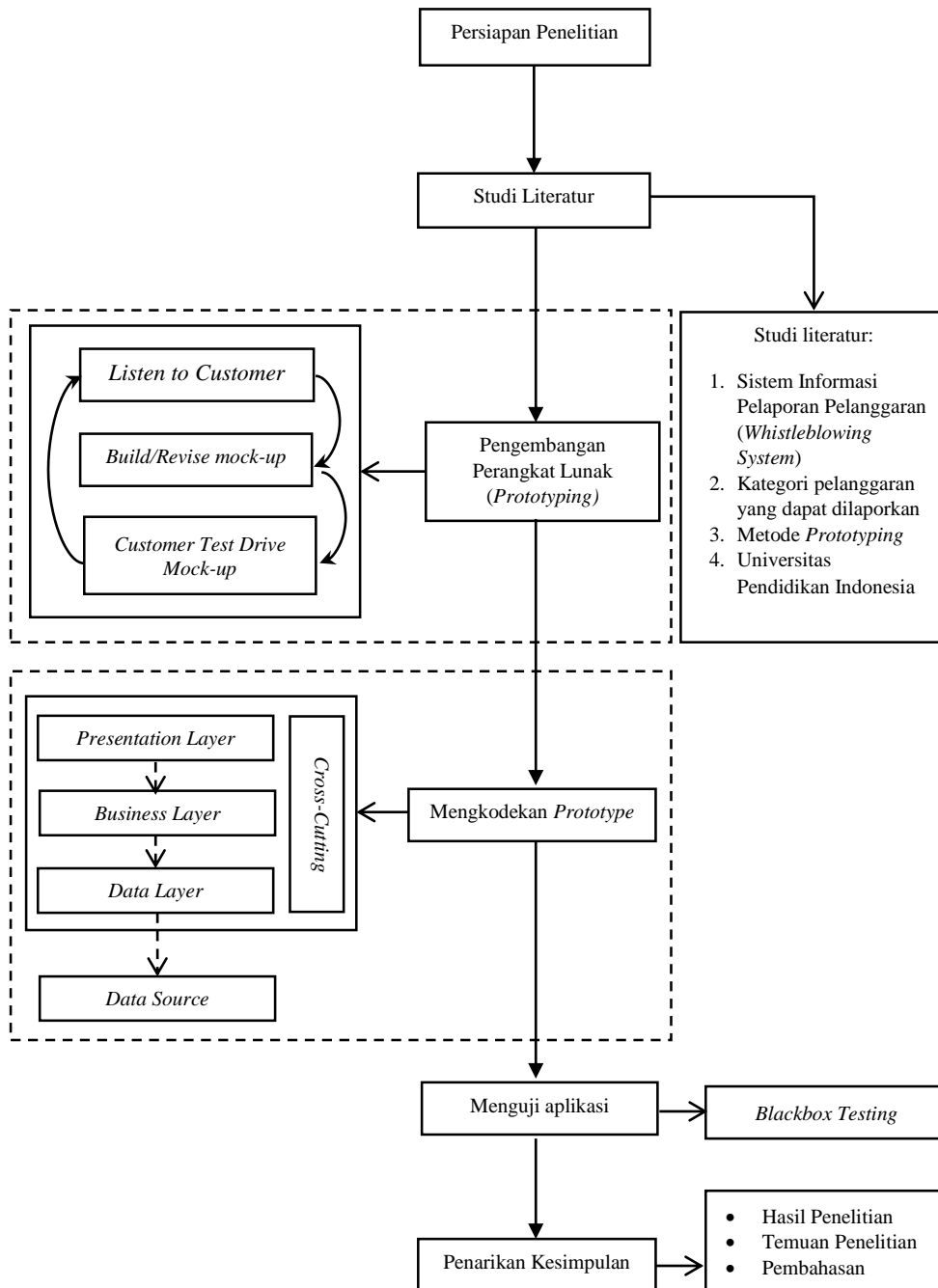


BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai metodologi penelitian, mulai dari desain penelitian, alat dan bahan penelitian, dan metodologi penelitian.

3.1 Desain Penelitian



Gambar 3.1 Desain Penelitian (Pressman, 2010; Team, 2009)

Penjelasan dari Gambar 3.1 adalah sebagai berikut:

1. Persiapan Penelitian

Tahap persiapan penelitian ini merupakan tahap awal dari penelitian, proses yang terjadi pada tahap ini yaitu dimulai dari identifikasi masalah yang akan dibahas, merumuskan masalah dan menentukan metode yang akan digunakan untuk menyelesaikan permasalahan tersebut.

2. Studi Literatur

Dalam tahap studi literatur ini peneliti mempelajari tentang sistem informasi *Whistleblowing*, kategori pelanggaran yang dapat dilaporkan didalam sistem informasi *Whistleblowing*, metode pengembangan aplikasi *Prototyping*, dan mempelajari tentang Universitas Pendidikan Indonesia. Dalam mempelajari tentang bahasan di atas, penulis mempelajari dari berbagai sumber seperti buku, jurnal, web terkait, dan sumber lainnya

3. Analisis Kebutuhan

Pada tahap ini, peneliti melakukan wawancara mengenai permasalahan yang sudah dirumuskan dan mengenai sistem informasi *whistleblowing* dengan *stakeholder*. Peneliti merumuskan tentang apa saja kebutuhan *user*, bagaimana sistem harus berfungsi.

4. Mengembangkan / Memperbaiki Prototipe

Setelah selesai melakukan analisis kebutuhan dengan *stakeholder*, peneliti membangun rancangan cepat dari sistem. Pembuatan rancangan cepat berdasarkan pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh para *end user*. Rancangan cepat akan dibuat dalam bentuk *use case diagram* dan *mockup*. Selesai membuat rancangan cepat, komunikasi dilakukan untuk mengetahui apakah rancangan cepat yang dibuat telah sesuai dengan kebutuhan dan keinginan *stakeholder*. Jika belum memenuhi maka rancangan cepat diperbaiki sampai memenuhi kebutuhan *stakeholder*.

5. Mengkodekan Aplikasi

Setelah rancangan cepat dinilai telah memenuhi kebutuhan *stakeholder*, peneliti melakukan pengkodean aplikasi menggunakan *software* Microsoft Visual Studio dengan bahasa pemrograman ASP.NET dengan *back end* menggunakan

bahasa pemrograman C# dan query sql untuk server database yang berupa sebuah *stored procedure* dalam SQL Server. Pengkodean aplikasi dilakukan dengan menerapkan *layered architecture* yang memisahkan aplikasi menjadi beberapa lapisan yang terdiri dari *presentation layer*, *business layer*, *service layer* dan *data layer*.

6. Menguji Aplikasi

Setelah pengkodean selesai dilakukan, sebelum mempresentasikan aplikasi yang telah dibuat aplikasi terlebih dahulu dilakukan pengujian seperti menggunakan *white box*, *black box*, *basis path*, pengujian arsitektur dan lain-lain. Jika aplikasi yang sudah dibangun belum sesuai dengan kebutuhan *stakeholder* maka dilakukan perbaikan pada tahap pengkodean aplikasi.

7. Hasil Penelitian, Temuan Penelitian dan Pembahasan.

Tahap ini membahas tentang hasil penelitian, temuan yang didapat serta pembahasan mengenai temuan penelitian.

8. Penarikan Kesimpulan

Tahap ini adalah tahap ketika proses pengembangan sistem selesai dilaksanakan sesuai dengan kebutuhan *stakeholder*. Selanjutnya menarik kesimpulan mengenai sistem yang telah dikembangkan dan di uji.

3.2 Metode Penelitian

Metode yang digunakan dalam penelitian ini terbagi menjadi dua bagian, yaitu metode pengumpulan data dan metode pengembangan perangkat lunak.

3.2.1 Metode Pengumpulan Data

Penulis mempelajari teori-teori yang menunjang dalam pelaksanaan penelitian yaitu teori-teori tentang sistem informasi, *whistleblowing system*, *good university governance* dan metode *prototyping* melalui buku-buku yang terkait, jurnal penelitian, artikel di internet dan sumber lainnya.

3.2.2 Metode Pengembangan Perangkat Lunak

Pengembangan sistem pada penelitian ini menggunakan model pengembangan *prototyping*. Tahapan pengembangan *prototyping* yaitu:

1. *Listen To Customer*

Pada tahap ini, yaitu tahap mendengarkan *customer* dimana berarti *developer* mendengarkan apa yang dibutuhkan dalam perangkat lunak yang akan dibangun menurut *customer*.

2. *Build/Revise Mock-up*

Pada tahap ini dilakukan pembangunan atau revisi *mock-up* sesuai dengan apa yang *customer* telah sampaikan pada tahap sebelumnya.

3. *Customer Test Drive Mock-up*

Pada tahap ini, *customer* mencoba atau mengetes *mock-up* yang telah dibuat atau direvisi pada tahap sebelumnya.

3.2.3 Arsitektur Sistem

Sistem yang dibangun akan menggunakan *layered architecture* yang merupakan arsitektur perangkat lunak yang membuat struktur sistem yang berlapis. *Layered architecture* berfokus pada pengelompokan fungsionalitas terkait dalam aplikasi ke dalam lapisan yang berbeda yang ditumpuk secara vertikal diatas satu sama lain. Fungsionalitas dalam setiap lapisan terkait dengan peran dan tanggung jawab bersama. Setiap lapisan pada arsitektur ini mempunyai peran serta kegunaan yang berbeda. Lapisan tersebut terdiri dari:

1. *Presentation Layer*

Lapisan ini bertanggung jawab untuk mengelola interaksi pengguna dengan sistem dan berkomunikasi dengan lapisan *business layer*.

2. *Business Layer*

Lapisan ini digunakan untuk mengolah dan mengalirkan data antara *presentation layer* dan *data Layer*. Lapisan ini merupakan lapisan inti dalam sebuah sistem yang berisi komponen serta *business logic* dari sistem.

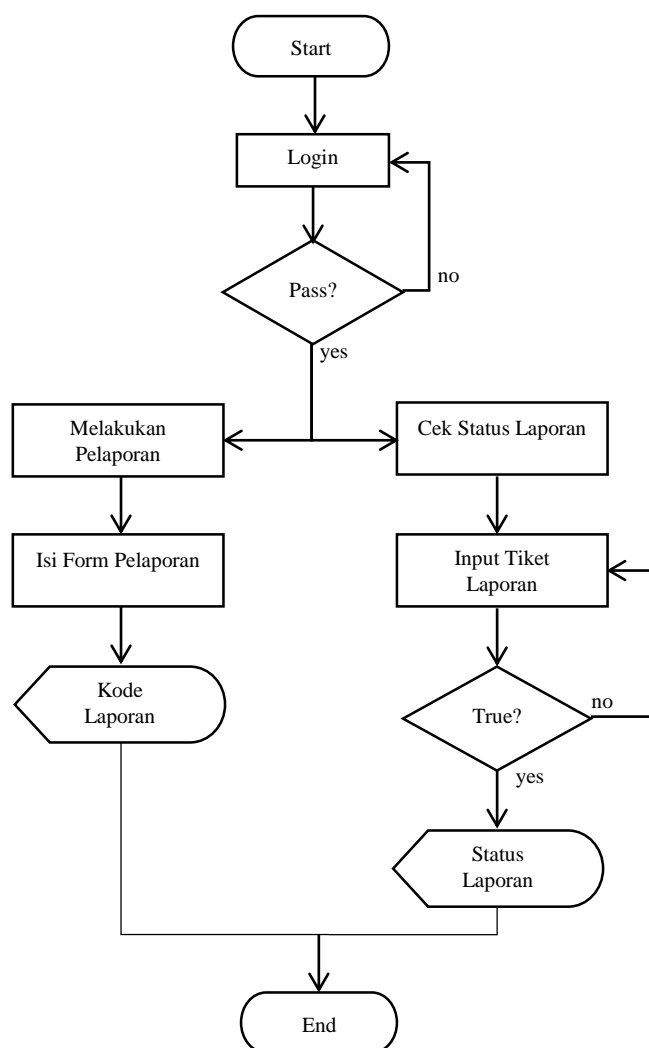
3. *Data Layer*

Class pada *data layer* digunakan untuk melakukan beberapa tindakan yang diperlukan dengan database seperti *create*, *update*, *insert* dan *delete*. *Class* tersebut berisi kode untuk koneksi terkait database seperti query SQL, stored procedure dan sebagainya.

Lapisan *cross-cutting* menyediakan *service* yang dapat digunakan semua lapisan. Salah satu lapisan *cross-cutting* adalah *security* yang menyediakan *class* untuk melakukan enkripsi dan deskripsi suatu data.

3.2.4 Flowchart Pelaporan

3.2.4.1 Flowchart Pelaporan oleh User

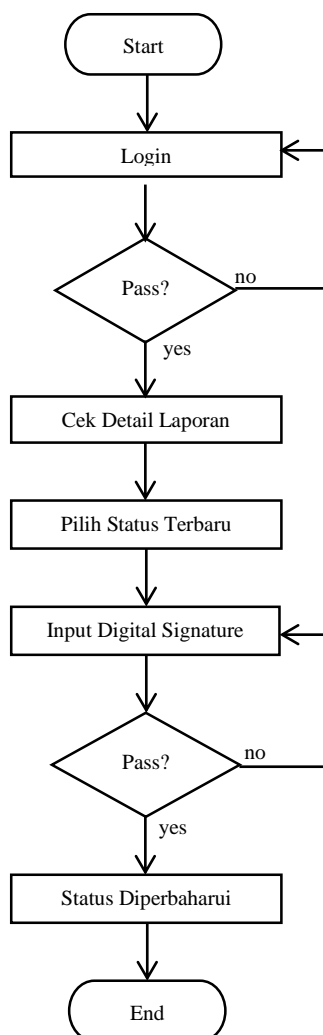


Gambar 3.2 Flowchat Pelaporan oleh User

Proses pelaporan pada Gambar 3.2 dilakukan oleh pelapor/*user* diasumsikan pelapor sudah memiliki akun di sistem pelaporan *whistleblowing*. Dimulai dari login lalu pelapor memilih untuk melakukan pelaporan atau melakukan pengecekan status laporan.

Ketika akan melakukan pelaporan, pelapor mendapat pilihan untuk menyembunyikan identitas atau tidak selanjutnya mengisi form pelaporan. Jika form sudah diisi dengan sesuai pelapor akan mendapatkan kode pelaporan yang dibutuhkan untuk melakukan pengecekan status laporan.

3.2.4.2 *Flowchart* Perubahan Status Laporan oleh Administrator



Gambar 3.3 Flowchart Perubahan Status Laporan oleh Administrator

Proses perubahan status laporan oleh administrator pada Gambar 3.3 dapat dilakukan oleh administrator dan super administrator yang mempunyai kendali penuh atas data laporan yang diterima oleh sistem. setelah melakukan proses login admin akan diarahkan menuju dashboard admin kemudian memilih laporan yang akan diperbaharui

statusnya. Setelah memilih status laporan, sistem akan memunculkan *digital signature* dengan cara sederhana yaitu memasukan password untuk meminimalisir penyalahgunaan akun admin. Jika berhasil maka sistem akan memperbaharui status laporan tersebut.

3.2.5 Penilaian Aplikasi

Pada tahap ini, aplikasi yang selesai dikembangkan akan dilakukan penilaian oleh ahli/*stakeholder*. Pembuatan kuesioner penilaian terdiri dari pertanyaan yang dibuat akan merujuk pada aspek-aspek *good university governance* yaitu transparansi, akuntabilitas, responsibilitas, independensi dan keadilan. Hasil pembuatan kuesioner terdapat pada tabel 3.1 berikut.

Tabel 3.1 Kuesioner Penilaian

| No | Indikator | Penilaian | | | | |
|-----------------|---|-----------|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Transparansi | | | | | | |
| 1. | Aplikasi yang dibangun memberikan informasi yang dapat diakses siapa saja. | | | | | |
| 2. | Aplikasi menyediakan informasi yang mudah dimengerti. | | | | | |
| 3. | Aplikasi dapat menjadi sebuah sistem untuk menjamin kualitas laporan keuangan dan pengungkapannya. | | | | | |
| Akuntabilitas | | | | | | |
| 4. | Aplikasi dapat membantu tugas pemeriksaan keuangan khususnya berkenaan dengan kecurangan yang terjadi. | | | | | |
| 5. | Aplikasi dapat digunakan sebagai metode untuk mengevaluasi kinerja staf pengajar, staf administrasi, dan staf manajerial. | | | | | |
| 6. | Aplikasi dapat membantu dalam pencegahan risiko terjadinya <i>fraud</i> . | | | | | |
| Responsibilitas | | | | | | |

| | | | | | | |
|--------------|--|------------------|----------|----------|----------|----------|
| 7. | Aplikasi dapat membantu satuan audit internal dalam menerima dan menanggapi isu-isu dan permasalahan-permasalahan yang terjadi. | | | | | |
| 8. | Aplikasi membantu Komite Audit lebih responsif terhadap permasalahan yang terjadi. | | | | | |
| No | Indikator | Penilaian | | | | |
| | | 1 | 2 | 3 | 4 | 5 |
| Independensi | | | | | | |
| 9. | Aplikasi membantu dalam menjaga pelaksanaan otonomi keuangan. | | | | | |
| Keadilan | | | | | | |
| 10. | Aplikasi memfasilitasi warga UPI dalam memenuhi hak dan kewajibannya dalam menegakan tata kelola universitas yang baik. | | | | | |
| 11. | Aplikasi membuat <i>stakeholder</i> merasa terlindungi dari upaya penyelewengan baik dalam bentuk usaha untuk kepentingan pribadi maupun benturan kepentingan. | | | | | |

Data yang diperoleh dari instrumen diolah dan dianalisis menggunakan metode rating scale dengan persamaan sebagai berikut (Sugiyono, 2012).

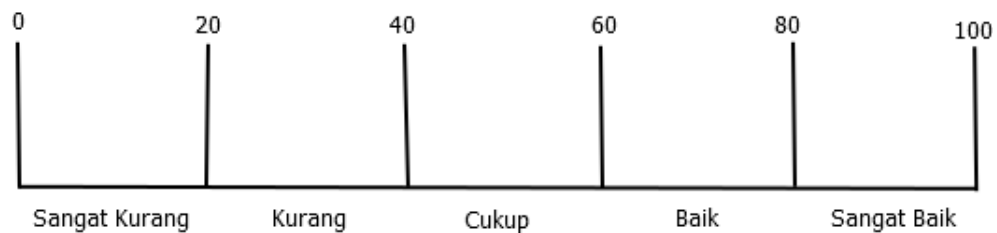
$$P = \frac{\text{skor hasil pengumpulan data}}{\text{skor ideal}} \times 100\%$$

Keterangan:

P : Angka Persentase

Skor Ideal : Skor tertinggi tiap butir x jumlah butir x jumlah responden

Selanjutnya hasil perhitungan menggunakan metode *rating scale* akan dikategorikan menjadi lima kategori dengan menggunakan skala sebagai berikut:



Gambar 3.4 Skala Rating Scale (Sugiyono, 2012)

Untuk memudahkan pemahaman kategori pada Gambar 3.4 akan direpresentasikan dalam tabel 3.2.

Tabel 3.2 Representasi Klasifikasi Kategori Berdasarkan *Rating Scale*.

| Skor Persentase (%) | Interpretasi |
|-------------------------|---------------|
| $P \leq 20 \%$ | Sangat Kurang |
| $20 \% < P \leq 40 \%$ | Kurang |
| $40 \% < P \leq 60 \%$ | Cukup |
| $60 \% < P \leq 80 \%$ | Baik |
| $80 \% < P \leq 100 \%$ | Sangat Baik |

Berdasarkan pada tabel 3.2 nilai P akan termasuk kategori sangat kurang apabila nilai P kurang dari atau sama dengan 20%, termasuk kategori kurang apabila nilai P lebih dari 20 % atau nilai P kurang dari atau sama dengan 40%, termasuk kategori cukup apabila nilai P lebih dari 40% atau nilai P kurang dari atau sama dengan 60 %, termasuk kategori baik apabila nilai P lebih dari 60% atau nilai P kurang dari atau sama dengan 80 %, termasuk kategori sangat baik apabila nilai P lebih dari 80% atau nilai P kurang dari atau sama dengan 100 %.

3.3 Alat dan Bahan Penelitian

3.3.1 Alat Penelitian

1. Perangkat keras (*Hardware*)
 - Prosesor : AMD A8
 - Memori (RAM) : 4 GB
 - Harddisk : 500GB
 - LAN Card : Ethernet Min. 100 MBps
 - Graphics : AMD Radeon R5 M330 DDR3L 2G

2. Perangkat lunak (*Software*)
 - Web Browser Mozilla Firefox
 - Microsoft Visual Studio 2013
 - SQL Server Management Studio 2012