

BAB III METODOLOGI

3.1 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan untuk mencapai suatu tujuan dalam penelitian. Metode pengumpulan data merupakan tahap awal dalam melakukan penelitian dengan mengumpulkan semua data dan informasi. Adapun metode pengumpulan data yang dilakukan pada penelitian antara lain:

a. Studi Pustaka

Tahap ini dilakukan dengan mencari sejumlah informasi dan referensi yang berhubungan dengan sistem informasi monitoring, metode perancangan dalam membangun sebuah sistem, dan metode pengujian perangkat lunak dengan *white box* testing. Studi pustaka yang dilakukan guna mendapatkan informasi yang berkaitan dengan judul penelitian, informasi-informasi tersebut diperoleh dari beberapa sumber seperti, buku, jurnal-jurnal penelitian, dan media internet.

b. Studi Lapangan (Observasi)

Tahap ini dilakukan dengan cara peninjauan pada *document* atau kumpulan laporan-laporan bulanan pelaksanaan kegiatan community tb-hiv care Aisyiyah. Adapaun pelaksanaan kegiatan yang dilakukan oleh *staff* data collection dan kepala SSR community TB-HIV care 'Aisyiyah Tanggamus yaitu mulai dari memantau kinerja setiap kader dilapangan yang melakukan penelusuran/penjaringan *suspect* (TB-HIV) di setiap daerah sesuai dengan daerah penempatan masing-masing, memantau pelaporan penjaringan *suspect* oleh kader di lapangan, mengelola atau mengkoleksi data capaian pelaksanaan kegiatan baik itu capaian indikator (seperti capaian *suspect* dan teridentifikasi tb-hiv) yang diperoleh dari penjaringan *suspect* di lapangan, melakukan pengawasan pada patient, dan menentukan reward atau honorarium oleh kader yang aktif dalam kegiatan penjaringan *suspect*.

c. Wawancara

Tahap ini dilakukan dengan cara bertatap muka bahkan kadang berkomunikasi melalui *whatsapp* dan setelah itu melakukan tanya jawab pada *staff* data collection Community TB-HIV Care 'Aisyiyah SSR Tanggamus, Lampung yaitu Bapak Afriadi Tanjung.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem merupakan sekumpulan langkah, prosedur, konsep, maupun aturan dalam mengembangkan dan membangun sebuah sistem. Dalam penelitian ini digunakan metode SDLC (*System Development Life Cycle*) dengan model proses air terjun (*waterfall*). Metode ini digunakan untuk memberikan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial, dengan harapan sistem yang dikembangkan dapat terselesaikan dan sesuai dengan yang diharapkan. Adapun tahapan-tahapan yang dilakukan dalam metode SDLC meliputi:

1. Analisa

Tahap ini bertujuan untuk mengetahui kebutuhan serta masalah yang harus di selesaikan. Analisis kebutuhan diperlukan untuk melakukan analisis kebutuhan perangkat lunak dan mengembangkan kebutuhan pengguna. Dalam pembuatan sistem informasi reporting ini diperlukan sekumpulan informasi yang dapat membantu dalam proses pembuatan sistem sehingga sistem dapat dirancang dengan baik agar sesuai dengan kebutuhan pengguna nantinya. Adapun prosedur yang dilakukan yaitu seperti mengumpulkan data, mengidentifikasi masalah, dan menganalisis kebutuhan fungsionalitas dan kebutuhan non-fungsionalitas pada sistem.

2. Tahap Desain / Perancangan

Tahap ini dilakukan berdasarkan pada hasil dari tahapan sebelumnya sebagai masukan sistem yang akan dikembangkan sesuai dengan kebutuhan pengguna. Proses desain dilakukan sebelum pembuatan kode program. Desain sistem bertujuan untuk memberikan gambaran yang jelas dalam pembuatan sistem, sehingga proses tahapan-tahapan yang dikerjakan dalam pengembangan atau pembuatan sistem berjalan sesuai dengan kebutuhan sistem. Perancangan sistem yang dilakukan yaitu mulai dari membuat *use case diagram*, *activity diagram*, membuat struktur tabel dan relasi tabel basis data dan perancangan desain antarmuka atau prototype.

3. Implementasi

Tahap ini dilakukan dengan mengimplementasi perancangan dengan kode program. Tahapan ini dikerjakan dengan mengkonversi desain ke dalam sistem informasi. Pembuatan kode program dikerjakan berdasarkan pada desain yang dirancang pada tahap sebelumnya. Pembuatan program menggunakan salah satu framework yaitu codeigniter. Setelah proses pembuat kode program selesai maka akan dilakukan pengujian pada sistem informasi yang dibangun yaitu sistem informasi reporting pada community tb-hiv care 'Aisyiyah Tanggamus Lampung.

4. Pengujain

Tahap ini dilakukan setelah sebuah sistem selesai dibuat, pengujian dilakukan menggunakan metode *white box testing*. Pengujian diperlukan untuk memeriksa *source code* pada program dan melakukan perbaikan apabila terdapat kesalahan (*bugs*) atau ketidaksesuaian pada sistem. Pengujian dengan *white box* dilakukan untuk memastikan bahwa semua kode pada program dilakukan eksekusi. Adapun teknik pengujian yang dipakai yaitu *statement coverage* dan *branch coverage*.

3.3 Metode Pengujian Sistem

Pengujian merupakan hal terpenting dalam sebuah perangkat lunak yang bertujuan untuk menemukan kesalahan ataupun kekurangan pada perangkat lunak yang akan diuji. Pada penelitian ini dilakukan pengujian perangkat lunak dengan menggunakan metode *white box testing*. Pengujian dengan metode *white box* dilakukan untuk meyakinkan semua perintah dan kondisi dieksekusi secara minimal. Adapun teknik yang digunakan pada metode *white box testing* yaitu *statement coverage testing* dan *branch coverage testing*.

3.3.1 Pengujian Statement Coverage

Pengujian dilakukan pada masing-masing *use case* yang ada. Adapun langkah-langkah pengujian yang dilakukan antara lain yaitu:

1. Menentukan proses mana yang akan dilakukan pengecekan pengujian (*Source Code*).
2. Merubah *source code* yang dipilih, kedalam bentuk *flowgraph*. *Flowgraph* merupakan alur dari logika program. Notasi pada *flowgraph* terdiri atas lingkaran dan panah. Lingkaran (*node*) menyatakan *statement procedural*. Sedangkan panah (*edge*) menyatakan aliran kendali atau alur perjalanan logika. Dengan bantuan *flowgraph* arah pengujian akan menjadi lebih mudah.
3. Menghitung nilai cc (*Cyclomatic Complexity*)

$$V(G) = E - N + 2$$
4. Penentuan *Independent Path*
Independent Path merupakan jalur pada program yang didapatkan berdasarkan perhitungan nilai *Cyclomatic Complexity* atau kemungkinan jalur yang diperoleh berdasarkan dari *flowgraph* yang dibuat.
5. Setelah memperoleh *independent path*, tahapan selanjutnya yaitu membuat pengujian dengan data tes untuk mengeksekusi jalur yang diperoleh untuk setiap *statement* yang di

uji coba minimal satu kali tes. Sebagai contoh Tabel 3.1 menunjukkan model *test case* untuk pengujian pada *use case* UC-01.

Tabel 3.1 *Test Case Statement* UC-01

Test Case Id	Test Case Name	Test Case Desc	Input Data	Expected Output

- Melakukan perhitungan nilai *coverage* pernyataan (baris kode) yang sudah dilakukan uji coba, dalam mencapai nilai 100% pada *statement coverage*, maka dilakukan perhitungan dengan rumus sebagai berikut:

$$\text{Statement coverage} = \frac{\text{Number of Statements Exercised}}{\text{Total Number of Statement}} \quad (3.1)$$

3.3.2 Pengujian Branch Coverage

Pengujian dilakukan pada masing-masing *use case* yang ada. Adapun langkah-langkah pengujian yang dilakukan antara lain yaitu:

- Menentukan proses mana yang akan dilakukan pengecekan pengujian (*Source Code*).
- Merubah *source code* yang dipilih, kedalam bentuk *flowgraph*. *Flowgraph* merupakan alur dari logika program. Notasi pada *flowgraph* terdiri atas lingkaran dan panah. Lingkaran (*node*) menyatakan *statement procedural*. Sedangkan panah (*edge*) menyatakan aliran kendali atau alur perjalanan logika. Dengan bantuan *flowgraph* arah pengujian akan menjadi lebih mudah.

- Menghitung nilai cc (*Cyclomatic Complexity*)

$$V(G) = E - N + 2$$

- Penentuan *Independent Path*

Independent Path merupakan jalur pada program yang didapatkan berdasarkan perhitungan nilai *Cyclomatic Complexity* atau kemungkinan jalur yang diperoleh berdasarkan dari *flowgraph* yang dibuat.

- Setelah memperoleh *independent path*, tahapan selanjutnya yaitu membuat pengujian dengan data tes untuk mengeksekusi jalur yang diperoleh untuk setiap keputusan cabang pada kondisi “true maupun false” dengan dilakukan uji coba minimal satu kali tes. Sebagai contoh Tabel 3.2 menunjukkan model *test case* untuk pengujian pada *use case* UC-01.

Tabel 3.2 *Test Case Branch UC-01*

Test Case Id	Test Case Name	Test Case Desc	Input Data	Expected Output

6. Melakukan perhitungan nilai *coverage* keputusan cabang yang sudah dilakukan uji coba, dalam mencapai nilai 100% pada *branch coverage*, maka dilakukan perhitungan dengan rumus sebagai berikut:

$$\text{Branch coverage} = \frac{\text{Number of Branch Exercised}}{\text{Total Number of Branch}} \quad (3.2)$$

3.4 Evaluasi

Pada tahap ini akan dilakukan evaluasi pada hasil akhir dalam membangun sistem dan melakukan pengujian pada Sistem Informasi Reporting pada Community TB-HIV Care Aisyiyah Tanggamus. Evaluasi dilakukan guna mengetahui seberapa baik penggunaan sistem dan perancangan sistem telah sesuai dengan yang diharapkan, seefektif apa pengujian menggunakan ke dua teknik tersebut yaitu *statement* dan *branch coverage* testing. Sehingga nantinya dapat memudahkan developer dalam mengembangkan sistem kedepan.