

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi. Biasanya suatu perusahaan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen.

Menurut (John F. Nash, 1995), Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang ditujukan untuk mengatur jaringan komunikasi yang penting, proses transaksi tertentu dan rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar untuk pengambilan keputusan yang tepat.

Menurut (Turban, McLean, dan Wetherbe, 1999), Sistem informasi adalah sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.

2.2 AppSheet

AppSheet adalah platform pengembangan tanpa kode yang memungkinkan siapa saja yang tidak memiliki pengalaman coding untuk mengembangkan aplikasi seluler dan web.

Aplikasi AppSheet dibuat dari sumber data seperti Google Spreadsheet, Excel, Cloud SQL, Salesforce, dan konektor serupa lainnya. Aktivitas pengguna aplikasi disinkronkan ke sumber data yang terhubung.

Aplikasi bersifat dinamis dan dapat digunakan di seluruh perangkat atau browser seluler. Desain antarmuka aplikasi menggunakan template UX untuk membuat peta, kalender, dasbor, dan lainnya. Alur kerja otomatis juga bisa disertakan ke dalam aplikasi untuk melakukan hal-hal seperti mengirim notifikasi, membuat email, membuat laporan kustom, dan memodifikasi data di seluruh sumber yang terhubung.

2.3 Tahapan Analisa

Setiap tahapannya yang digunakan dalam pengerjaan tahap berikutnya. Pendokumentasian pada setiap keluaran ditujukan untuk membantu proses pengembangan. Banyak literasi yang terjadi pada saat melakukan domain model, analisis use case, dan sebagainya. Model statis yang dihasilkan terus diperbaiki secara bertahap dengan bantuan model dinamis yaitu use case, robustness analysis dan sequence diagram. Enterprise Architect memiliki 6 tahapan, yaitu: Requirement, Analysis / Preliminary Design, Milestone 2: Preliminary Design Review, Detailed Design, Milestone 3: Critical Design Review, dan Implementation. Namun pada penelitian ini belum sampai dilakukannya tahap implementasi.

Tabel 2.1 Artefak Metode Process

Phase	Artifacts	UML
<i>Requirement</i>	<i>Functional Requirement</i>	No
	<i>Domain Modelling</i>	Yes
	<i>Behavioral Requirement</i>	Yes
	<i>Requirement Review</i>	Yes
<i>Analysis/Preliminary Design</i>	<i>Robustness Analysis</i>	Yes

<i>Milestone 2: Preliminary Design Review</i>	Domain Model	Yes
<i>Detailed Design</i>	<i>Sequence Diagram</i>	Yes
	<i>Class Diagram</i>	Yes
<i>Milestone 3: Critical Design Review</i>	<i>Consistency Analysis</i>	Yes
	<i>Correctness</i>	No

2.3.1 Fase Requirement

a) Functional Requirement

Pada tahap ini penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Selain itu kadang - kadang juga secara jelas menentukan apa yang tidak dikerjakan oleh sistem. Functional requirement menggambarkan system requirement secara detail seperti input, output dan pengecualian yang berlaku.


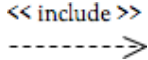
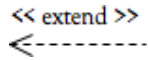



b) Domain Modelling

Domain Modelling merupakan teknik pengidentifikasian object- object pada kata benda yang terdapat pada daftar requirement yang diklasifikasikan pada area (domain) permasalahan yang sama untuk dijadikan candidate class pada class diagram.

c) Behavioral Requirement

Dilakukan pembangunan GUI *storyboard* atau rancangan tampilan awal serta *use case diagram* yang menggambarkan fitur dan fungsi sistem. Berikut simbol-simbol yang digunakan dalam *use case diagram*.

Tabel 2.2 Simbol Use Case Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Merupakan peran orang, sistem lain, atau alat ketika berhubungan dengan use case
	<i>Include</i>	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	<i>Extend</i>	Merupakan perluasan dari use case lain jika kondisi atau Syarat terpenuhi
	<i>Association</i>	Abstraksi dari penghubung antara actor dengan use case
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	<i>Use case</i>	Asbtraksi dari penghubung antara actor dengan use case

d) Milestone 1: Requirement Review

Pada tahap ini dilakukan peninjauan terhadap analisis sebelumnya. Hasil yang didapatkan adalah sedikit penambahan dan pembaharuan pada *functional requirement* dan pada GUI *storyboard*.

2.3.2 Analysis/Preliminary Design

a) Robustness Analysis

Pada tahap ini dilakukan penelitian menggunakan *Robustness diagram* yang merupakan jembatan/penghubung antara analisis dan perancangan sistem. Diagram ini merupakan gambaran objek dari suatu use case, yang tujuannya untuk menyempurnakan use case dan model objek.

2.3.3 Milestone 2: Preliminary Design Review


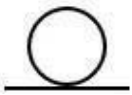
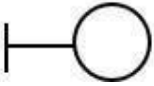



Pada tahap ini dilakukan pembaharuan terhadap *domain model* berdasarkan pada *use case*, *use case scenario* dan *robustness analysis* yang telah dibuat terdapat data yang dimasukkan atau dihapus dari *domain model*.

2.3.4 Detailed Design

a) Sequence Diagram

Sequence diagram dibuat sesuai *use case* dan *robustness diagram*. Penamaan untuk *actor*, *boundary*, *control*, dan *entity* yang dipakai merupakan istilah yang disesuaikan dengan *domain model* yang telah dibuat. Alur *sequence diagram* mengikuti *robustness diagram* yang sebelumnya mengikuti *use case scenario*. Berikut simbol-simbol yang digunakan dalam *sequence diagram*.

Tabel 2.3 Simbol Sequence Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Digunakan untuk menggambarkan user pengguna
	<i>Entity Class</i>	Digambarkan untuk menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Digunakan untuk menggambarkan sebuah form
	<i>Control Class</i>	Bertanggung jawab dalam mengatur kelas-kelas yang lain.
	<i>Life Line</i>	Object entity, antarmuka yang saling berinteraksi
	<i>Message</i>	Spresifikasi dan komunikasi antar objek yang memuat infromasi-informasi tentang aktifitas yang terjadi

b) Class Diagram

Class diagram dibuat dari perkembangan *domain model* dan diagram-diagram sebelumnya. Terdapat *entity, control, model, dan view*.

2.3.5 Milestone 3: Critical Design Review

Pada tahap ini dilakukan evaluasi dengan tujuan memastikan perancangan sistem yang telah dibuat sesuai dengan analisis dan kebutuhan yang telah dijabarkan sebelumnya.