

BAB II

LANDASAN TEORI

2.1 Kajian Pustaka

Sofyan dan Iryanti, (2018) dalam penelitiannya membuat aplikasi penanganan keluhan mahasiswa berbasis *website* dengan menggunakan prototipe, objek penelitian di Institut Teknologi Purwokerto. Aplikasi penanganan keluhan ini diperlukan untuk menangani pengaduan dari mahasiswa terhadap kampus berdasarkan pada bidang kerja staff. Perancangan aplikasi sendiri menggunakan metode prototipe, metode prototipe dipilih karena dalam tahapan prosesnya singkat dan mudah dipahami oleh pengguna aplikasi. Hasil dari penelitian ini bahwa metode prototipe ini baik digunakan dalam pembuatan perancangan sebuah sistem maupun aplikasi berbasis *web*.

Nalendro (2019) dalam penelitiannya merancang dan mengimplementasikan sebuah *system* informasi yang dapat menampung semua keluhan mengenai fasilitas dan pelayanan yang disediakan oleh universitas. Adapun pihak pengelola universitas akan mendapatkan notifikasi bila ada keluhan yang masuk sehingga bisa segera menindaklanjuti keluhan tersebut. Sistem ini juga akan menyajikan informasi statistik keluhan yang masuk agar dapat menjadi dasar bagi seorang pimpinan universitas untuk membuat suatu kebijakan.

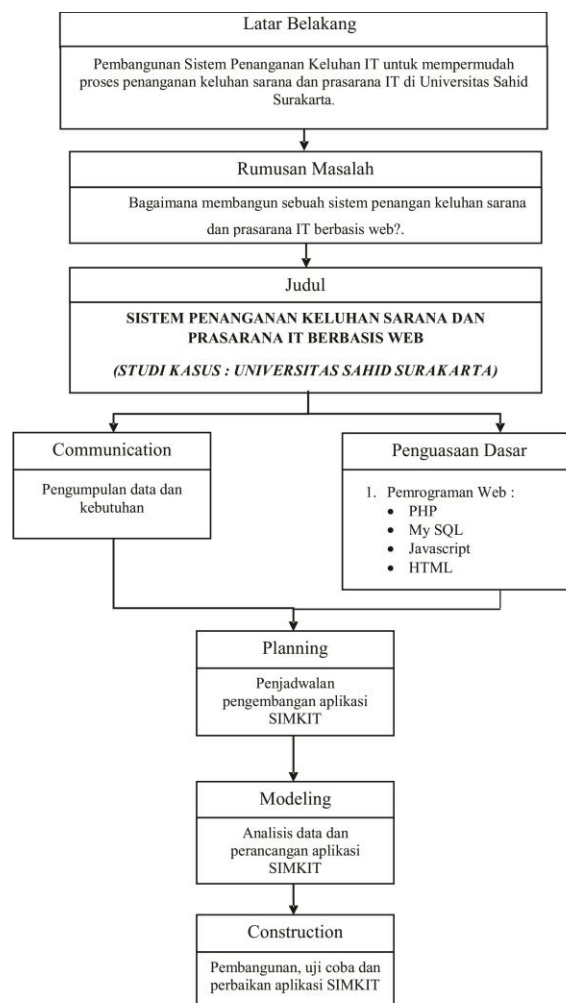
Melani (2019) dalam penelitiannya membuat sistem pengaduan layanan akademik bagi mahasiswa menggunakan *responsive web design* mampu mempermudah mahasiswa dalam menyampaikan keluhan terhadap layanan akademik yang diterima. Sistem ini terdiri dari tiga menu yaitu layanan BAAK, layanan keuangan dan layanan Sarpras dimana mahasiswa dapat memberikan keluhannya terhadap masing-masing layanan. Print out dari keluhan mahasiswa ini adalah laporan yang akan diberikan kepada pimpinan untuk ditindak lanjuti.

Khusnuliawati, dkk. (2018) dalam penelitiannya membuat perancangan sistem informasi akademik berorientasi objek menggunakan *unified modelling language*. Perancangan perangkat lunak berorientasi objek cocok digunakan untuk

merancang sistem yang kompleks termasuk sistem informasi akademik di Universitas Sahid Surakarta. Hal itu dijadikan acuan untuk membuat program pada penelitian ini sebagai dasar perancangan *website* yang sama-sama menggunakan *unified modelling language*.

2.2 Kerangka Pemikiran

Alur pemikiran penelitian perancangan dan pembangunan Sistem Penanganan Keluhan Sarana dan Prasarana IT pada laporan ini ditunjukkan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran Penelitian

2.3 Teori Pendukung

2.3.1 Manajemen Keluhan

Manajemen keluhan adalah tentang menangani dan menyelesaikan keluhan individu. Ini juga tentang menggunakan informasi untuk mengidentifikasi potensi perbaikan pada pengambilan keputusan, praktik, dan pemberian layanan. Manajemen keluhan merupakan bagian integral dari layanan pelanggan yang berkualitas dan memberikan manfaat nyata bagi agensi, staf, pelanggan, dan orang-orang di komunitas (Barsalou, 2020).

2.3.2 Sarana dan Prasarana

Sarana adalah alat langsung untuk mencapai tujuan pendidikan, misalnya ruangan, buku, perpustakaan, laboratorium, dan sebagainya. Sarana pendidikan antara lain bangunan, dan perabotan sekolah, alat pelajaran yang terdiri atas pembukuan alat-alat peraga dan laboratorium, serta media pendidikan yang dapat dikelompokkan menjadi audiovisual yang menggunakan alat terampil. Sedangkan prasarana adalah alat tidak langsung untuk mencapai tujuan pendidikan, seperti : lokasi atau tempat, bangunan sekolah, lapangan olahraga dan sebagainya (Kasan, 2007) yang dikutip dalam (Nurabadi, 2014).

2.3.3 Teknologi Informasi

Teknologi informasi, atau dalam bahasa Inggris dikenal dengan istilah *information technology (IT)*. Menurut McKeown dalam (Suyanto, 2005) teknologi informasi merujuk pada seluruh bentuk teknologi yang digunakan untuk menciptakan, menyimpan, mengubah, dan menggunakan informasi dalam segala teknologi informasi merupakan sebuah bentuk umum yang menggambarkan setiap teknologi yang membantu menghasilkan, memanipulasi, menyimpan, mengkomunikasikan, dan atau menyampaikan informasi.

2.3.4 Unified Modelling Language

Unified Modelling Language (UML) adalah ketentuan umum teknik menggambar diagram dan istilah-istilah dalam *object-oriented* yang dapat digunakan untuk memodelkan suatu proyek pengembangan sistem dari proses analisis sampai dengan proses implementasi (Dennis, dkk., 2015). UML diresmikan oleh Object Management Group (OMG) pada tahun 1997 dan telah mengalami

beberapa revisi sampai dengan versi terakhirnya saat ini yaitu UML 2.5. UML diagram terdiri dari 15 jenis diagram yang terbagi menjadi 2 kelompok: *structure diagram* dan *behavior diagram*.

Structure diagram adalah diagram untuk merepresentasikan data dan relasi statis dalam suatu sistem informasi (Dennis, dkk., 2015). *Structure diagram* yang akan digunakan untuk merepresentasikan sistem dalam laporan ini antara lain: *class diagram* dan *deployment diagram*.



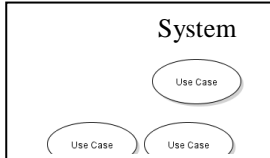
Behavior diagram adalah diagram untuk menggambarkan relasi dinamis antar objek yang merepresentasikan bisnis yang terjadi dalam sistem informasi (Dennis, dkk., 2015). *Behavior diagram* yang akan digunakan dalam laporan ini antara lain: *use-case diagram*, *activity diagram*, dan *sequence diagram*.

2.3.4.1 Use-Case Diagram


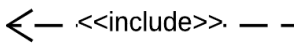
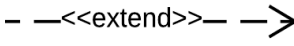
Use-case diagram adalah diagram yang menggambarkan fungsi utama suatu sistem dengan sederhana dan pengguna yang akan berinteraksi dengan sistem tersebut (Dennis, dkk., 2015). Elemen-elemen yang digunakan dalam pembuatan *use-case diagram* umumnya terdiri dari *actor*, *use-case*, dan *subject boundary* seperti yang ditunjukkan pada Tabel 2.1.

Use-case diagram juga notasi relasi untuk menggambarkan interaksi antar elemen. Interaksi tersebut dapat terjadi antara *actor* dengan *use-case* atau *use-case* dengan *use-case* lainnya. Beberapa relasi yang umum digunakan dalam *use-case diagram* dapat dilihat pada Tabel 2.2

Tabel 2.1 Elemen-Elemen *Use-Case Diagram*

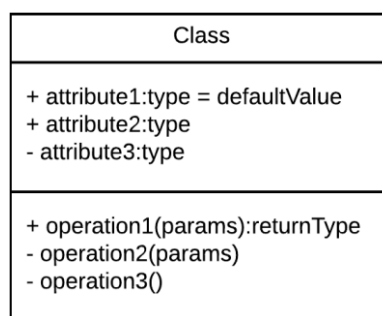
No.	Elemen/Keterangan	Notasi
1	Aktor: jenis peran pengguna dalam sistem.	
2	Use-case: proses pokok yang dilakukan sistem sehingga dapat bermanfaat bagi aktor.	
3	Subject Boundary: menjelaskan cakupan sistem dan menggambarkan bagian mana yang merupakan internal sistem atau eksternal sistem.	

Tabel 2.2 Jenis Relasi dalam *Use-Case Diagram*

No.	Relasi/Keterangan	Notasi
1	Association Relationship: penghubung yang menunjukkan interaksi antara aktor dengan <i>use-case</i> .	
2	Include Relationship: menggambarkan penyertaan fungsi suatu <i>use-case</i> ke dalam <i>use-case</i> lain.	
3	Extend Relationship: menggambarkan fungsi tambahan bersifat opsional dari suatu <i>use-case</i> .	

2.3.4.2 *Class Diagram*

Class diagram adalah model statis yang menunjukkan atribut kelas, operasi kelas, dan relasi antar kelas yang konstan selama berada di dalam sistem (Dennis, dkk., 2015). Pada umumnya *class diagram* hanya terdiri dari elemen utamanya yaitu kelas (*class*). Kelas berfungsi untuk menyimpan dan mengelola informasi dalam suatu sistem. Kelas dinotasikan dalam bentuk persegi seperti yang terlihat pada Gambar 2.2.



Gambar 2.2 Kelas dalam *Class Diagram*

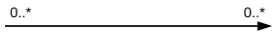
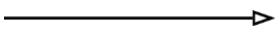
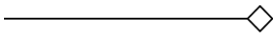

Gambar 2.2 menunjukkan bahwa kelas terbagi menjadi 3 bagian. Tiga komponen kelas tersebut adalah *class name* (atas), *attribute* (tengah), dan *operation* (bawah). Selain itu, setiap *attribute* dan *operation* memiliki tingkat akses yang menentukan perilaku masing-masing *attribute* atau *operation* terhadap kelas lain. Keterangan dari masing-masing komponen dan *visibility* dalam suatu kelas dapat dilihat pada Tabel 2.3.

Class diagram menunjukkan relasi antar kelas satu dengan kelas yang lain. Ada beberapa macam relasi yang umum digunakan dalam *class diagram*, antara lain: *association*, *generalization*, *aggregation*, dan *composition*. Keterangan dari jenis-jenis relasi dalam *class diagram* dapat dilihat pada Tabel 2.4. Relasi antar kelas juga memiliki *multiplicity* yang merupakan penulisan tentang bagaimana suatu objek berelasi dengan dengan objek lain (Dennis, dkk., 2015). *Multiplicity* direpresentasikan dengan angka pada garis relasi antar kelas dengan format *angka minimum..angka maksimum* seperti pada Tabel 2.5.

Tabel 2.3 Kelas dalam *Class Diagram*

No.	Komponen	Keterangan
1	<i>Class Name</i>	<i>Identifier</i> unik suatu kelas.
2	<i>Attribute</i>	Properti dari kelas yang menyimpan informasi-informasi dari kelas itu sendiri.
3	<i>Operation</i>	Aksi atau fungsi yang dapat dilakukan oleh kelas. Dapat memiliki parameter yang dibutuhkan operasi, ditunjukkan dalam tanda kurung setelah nama operasi.
4	<i>Visibility</i>	Tingkat akses <i>attribute</i> atau <i>operation</i> dalam kelas. Visibilitiy dapat dibagi menjadi tiga jenis, yaitu: <ol style="list-style-type: none"> 1. <i>Public</i> (+), dapat diakses oleh kelas lain. 2. <i>Protected</i> (#), hanya dapat diakses oleh kelas itu sendiri atau subkelasnya. 3. <i>Private</i> (-), hanya dapat diakses oleh kelas itu sendiri.

Tabel 2.4 Relasi Antar Kelas dalam *Class Diagram*

No.	Relasi	Keterangan	Notasi
1	<i>Association</i>	Merepresentasikan hubungan antar kelas. Dilabeli dengan kata kerja dan memiliki <i>multiplicity</i> .	
2	<i>Generalization</i>	Merepresentasikan hubungan pewarisan antara <i>superclass</i> dengan <i>subclass</i> .	
3	<i>Aggregation</i>	Merepresentasikan hubungan kelas yang merupakan bagian dari kelas lain.	
4	<i>Composition</i>	Merepresentasikan hubungan satu kesatuan antara satu kelas dengan kelas yang lain.	







Tabel 2.5 *Multiplicity* pada Relasi *Class Diagram*

No.	Multiplicity	Keterangan
1	1	Satu dan hanya satu.
2	0..*	Boleh tidak ada atau 1 atau lebih.
3	1..*	1 atau lebih.
4	0..1	Boleh tidak ada, maksimal 1.
5	<i>n..m</i>	Batasan antara <i>n</i> sampai dengan <i>m</i> . Contoh 2..4 mempunyai arti minimal 2 maksimal 4.

2.3.4.3 *Activity Diagram*

Activity diagram adalah diagram yang menggambarkan aktivitas primer dan relasi antar aktivitas dalam suatu proses (Dennis, dkk., 2015). Tabel 2.6 adalah elemen-elemen yang digunakan dalam pembuatan *activity diagram*.

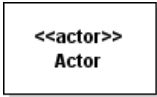
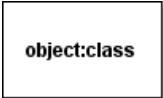


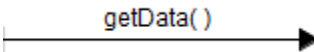
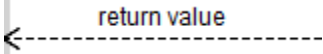
Tabel 2.6 Elemen-Elemen *Activity Diagram*

No.	Elemen/Keterangan	Notasi
1	Action: satu kegiatan yang tidak dapat diuraikan. Activity: kumpulan dari aksi-aksi.	
2	Control Flow: menunjukkan urutan eksekusi.	
3	Initial Node: titik awal aktivitas (aksi).	
4	Final Activity Node: titik berhenti semua aliran objek atau aktivitas (aksi).	
5	Decision Node: menentukan jalur mana yang akan ditempuh setelah dilakukan tes kondisi.	
6	Fork Node: membagi aktivitas menjadi kegiatan-kegiatan paralel.	

2.3.4.4 *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan objek-objek yang ikut menjadi bagian dari satu *use-case* dan pesan-pesan yang melewatinya dari waktu ke waktu (Dennis, dkk., 2015). Tabel 2.7 adalah elemen-elemen yang digunakan dalam pembuatan *sequence diagram*.

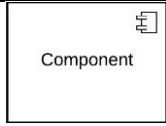




Tabel 2.7 Elemen-Elemen *Sequence Diagram*

No.	Elemen/Keterangan	Notasi
1	Actor: eksternal sistem berupa orang atau sistem lain yang mendapatkan manfaat dari <i>use-case</i> .	
2	Object: sesuatu yang ikut dalam rangkaian kegiatan dengan mengirim atau menerima pesan.	
3	Lifeline: menunjukkan waktu hidup suatu objek dalam rangkaian kegiatan.	
4	Execution Occurrence: menandakan ketika suatu objek mengirim atau menerima pesan.	
5	Request Message: permintaan informasi dari satu objek ke objek lain.	
6	Return Message: menjawab permintaan informasi yang dikirim oleh suatu objek.	

2.3.4.5 *Component Diagram*

Component diagram adalah diagram yang digunakan untuk mengilustrasikan hubungan fisik antar komponen perangkat lunak dalam sistem (Dennis, dkk., 2015). *Component diagram* dapat digunakan untuk pemodelan *source code*, *executable release*, dan *database*. Elemen-elemen yang digunakan dalam pembuatan *component diagram* dapat dilihat pada Tabel 2.8.

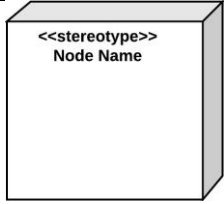

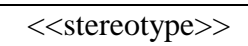
Tabel 2.8 Elemen-Elemen *Component Diagram*

No.	Elemen/Keterangan	Notasi
1	Component: entitas yang diperlukan untuk mengeksekusi suatu fungsi. Sebuah <i>component</i> menyediakan dan membutuhkan suatu kegiatan lain dari <i>interface</i> atau <i>component</i> lain.	
2	Provided Interface: menggambarkan informasi yang dihasilkan suatu <i>component</i> yang akan digunakan oleh <i>component</i> lain.	
3	Required Interface: menggambarkan kebutuhan informasi atau data suatu <i>component</i> untuk menjalankan fungsinya.	
4	Ports: batas interaksi antara entitas dengan lingkungannya. Akses yang terjadi tidak langsung pada entitas tersebut, tetapi akses dilakukan pada entitas di dalamnya.	
5	Package: pengelompokan elemen dalam suatu sistem.	

2.3.4.6 *Deployment Diagram*

Deployment diagram adalah diagram yang digunakan untuk merepresentasikan hubungan antara komponen perangkat keras yang digunakan dalam infrastruktur fisik dari sistem informasi (Dennis, dkk., 2015). *Deployment diagram* juga dapat digunakan untuk merepresentasikan komponen perangkat lunak yang terpasang dalam perangkat keras dari suatu sistem informasi (Dennis, dkk., 2015). Tabel 2.9 adalah elemen dan relasi yang akan digunakan dalam pembuatan *deployment diagram*.

Tabel 2.9 Elemen-Elemen *Deployment Diagram*

No.	Elemen/Keterangan	Notasi
1	Device Node: perangkat keras yang dibutuhkan di dalam desain arsitektur fisik sistem. Execution Environment Node: sistem komputer atau perangkat lunak yang berada di dalam <i>device node</i> .	
2	Artifact: sistem informasi (perangkat lunak) yang akan dipasang pada arsitektur fisik (perangkat keras).	
3	Communication Path: merepresentasikan hubungan antar <i>node</i> .	

2.3.5 Web

Pressman (2014) dalam bukunya menjelaskan bahwa *web* adalah salah satu kategori perangkat lunak yang berbasis jaringan (*network-centric*) yang mencakup berbagai macam aplikasi dari bentuk yang paling sederhana berupa kumpulan arsip *hypertext* sampai dengan yang kompleks berupa lingkungan komputasi canggih yang terintegrasi dengan *database* dan aplikasi bisnis. Pengguna dapat mengakses aplikasi *web* melalui *web browser* yang terhubung dalam suatu jaringan.

2.3.6 Metode Pengujian *Webqual*

Webqual 4.0 disusun berdasarkan tiga area utama yaitu kualitas informasi, kualitas interaksi dan usability. Barnes dan Vidgen, (2003) mendefinisikannya sebagai berikut: Kualitas informasi meliputi informasi yang akurat, informasi yang dapat dipercaya, informasi yang up to date, informasi sesuai dengan topik bahasan, kemudahan informasi untuk dimengerti, kedetailan informasi dan informasi yang disajikan dalam format desain yang sesuai. Kualitas interaksi meliputi kemampuan memberikan rasa aman saat transaksi, memiliki reputasi yang bagus, memudahkan komunikasi, menciptakan perasaan emosional yang lebih personal, memiliki kepercayaan dalam memberikan informasi pribadi, mampu menciptakan komunitas yang spesifik, memberikan keyakinan bahwa janji yang disampaikan akan ditepati. Kualitas usability meliputi kemudahan *website* untuk dipelajari, kemudahan untuk dimengerti, kemudahan untuk ditelusuri, kemudahan untuk digunakan, kemenarikan *website, interface* yang menyenangkan, memiliki kompetensi yang baik dan memberikan pengalaman baru yang menyenangkan.

Selain itu dimensi dari *webqual* 4.0 berasal dari banyak System Information Assesment, ini berarti *Webqual* mempunyai fondasi yang kuat seperti Servqual Parasuraman atau user satisfaction model oleh Bailey. Berikut adalah tabel pertanyaan dari *Webqual* beserta sumber pendukung pertanyaan dari *Webqual* 4.0 dapat dilihat pada Tabel 2.10.

Tabel 2.10 *Webqual 4.0* instrumen.

Dimensi	Kuesioner <i>Webqual 4.0</i>
<i>Usability</i>	<i>I find the site easy to learn to operate</i>
	<i>My interaction with the site is clear and understandable</i>
	<i>I find the site easy to navigate</i>
	<i>I find the site easy to use</i>
	<i>The site has an attractive appearance</i>
	<i>The design is appropriate to the type of site</i>
	<i>The site conveys a sense of competency</i>
	<i>The site creates a positive experience</i>
<i>Information Quality</i>	<i>Provides accurate information</i>
	<i>Provides believable information</i>
	<i>Provides timely information</i>
	<i>Provides relevant information</i>
	<i>Provides easy to understand information</i>
	<i>Provides information at the right level of detail</i>
	<i>Present the information in appropriate format</i>
<i>Service Interaction</i>	<i>Has a good reputation</i>
	<i>It feels safe to complete transaction</i>
	<i>My personal information feels secure</i>
	<i>Creates a sense of personalization</i>
	<i>Convey a sense of community</i>
	<i>Make it easy to communicate with the organization</i>
	<i>I feel confident that goods/services will be delivered as promised</i>
<i>Overall</i>	<i>Overall View of The Website</i>

2.3.7 Skala Linkert

Angket atau disebut juga *questionnaire* adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon, sesuai dengan permintaan pengguna. Tujuan penyebaran angket adalah mencari informasi dari responden tanpa khawatir bila responden memberikan jawaban yang tidak sesuai dengan kenyataan (Riduwan, 2005). Dalam penelitian ini, angket dibutuhkan untuk mengukur tingkat kelayakan penggunaan aplikasi.

Menurut Riduwan (2005), skala sikap dibagi menjadi lima bentuk, yaitu Skala Likert, Skala Guttman, Skala Defferensial Simantict, Rating Scale dan Skala Thurstone. Skala Likert adalah sebuah skala psikometri yang digunakan dalam kuesioner untuk mendapatkan pilihan peserta dan tingkat persetujuan dengan suatu pernyataan. Responden diminta untuk menunjukkan tingkat persetujuan pada pernyataan yang diberikan dengan cara memilih skala yang ordinal. Skala yang

paling sering digunakan adalah skala dengan 5 pilihan. Mulai dari "Sangat Tidak Setuju" sampai "Sangat Setuju" dan terdapat pilihan "Netral" di tengah-tengah (Saputra dan Nugroho, 2017) (Emerson, 2017). Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok tentang kejadian atau gejala sosial. Pengukuran sikap, pendapat dan persepsi seseorang harus melalui proses pengolahan data, Angket yang sebelumnya telah diisi kemudian direkapitulasi sehingga dapat dilakukan perhitungan skor.

Untuk menemukan range untuk setiap nilai, dapat ditentukan dengan persamaan sebagai berikut :

$$\text{Range} = \frac{\text{Nilai Skala} \times \text{Jumlah Responden}}{\text{Nilai Tertinggi} \times \text{Jumlah Responden}} \times 100\%$$