

BAB II

TINJAUAN PUSTAKA

2.1. Reservasi

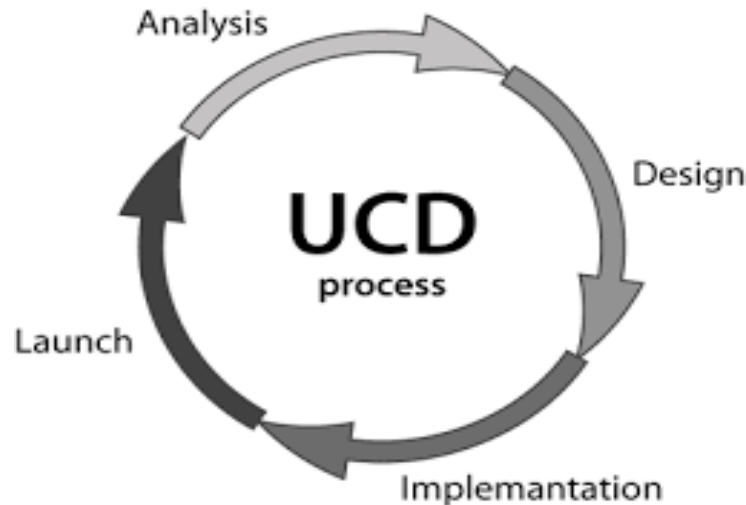
Reservasi adalah sebuah proses klerikal atau elektronik dimana produk perjalanan seperti tiket pesawat, kamar hotel, kamar pada kapal pesiar tersedia untuk dipakai dan pada akhirnya dibeli oleh individu secara spesifik (Monaghan, 2017).

2.2. Tiket

Menurut (Hendi, 2016), tiket adalah sebuah karcis yang digunakan dalam suatu organisasi untuk melacak deteksi, pelaporan, dan resolusi dari beberapa jenis masalah. Trouble sistem ticketing berasal dari manufaktur sebuah dasar sistem pelaporan kertas. Sekarang kebanyakan berbasis web dan terkait dengan hubungan manajemen pelanggan seperti *call center* atau *e-business*.

2.3. User Centered Design (UCD)

UCD (*User Centered Design*) adalah sebuah pendekatan pembangunan perangkat lunak yang memusatkan fokus pada pengguna pada setiap proses pembangunan perangkat lunak (Preece, 2002). Pada UCD perancangan harus terkoneksi langsung dengan user/pengguna akhir atau calon user/pengguna melalui beberapa metode antara lain: wawancara, survey dan workshop pada saat perancangan. Pendekatan UCD dapat digunakan untuk meningkatkan usability dari produk. Akan tetapi, secara praktis penerapan UCD seringkali menemui berbagai kendala yang kurang memenuhi pencapaian user experience yang maksimal, kurang melibatkan end-to-end user pada tahap pengembangan, serta kurang menelusuri kepuasan pelanggan (Mao, Karel, Paul, Tom, 2015).



Gambar 2.1 Model UCD

Pada gambar 2.1 tersebut, Eason menggambarkan empat langkah utama dalam pengembangan, yakni perencanaan, perancangan, implementasi dan pengelolaan sistem. Tahap empat pendekatan yang bisa dijelaskan dari gambar diatas adalah :

1. Soft System Methodology (SSM) SSM lebih fokus pada perencanaan.
2. Open Task Analysis (OSTA), OSTA lebih fokus pada langkah awal perancangan.
3. Multiview Metodologi yang lengkap dengan rentang perencanaan sampai dengan implementasi.
4. Star Life Cycle fokus utama pada perancangan.

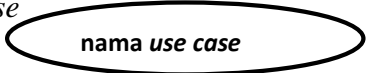
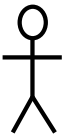



2.4. UML


Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Rosa & Shalahuddin, 2018). Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.4.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa & Shalahuddin, 2018). simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini:

Tabel 2.1 Simbol Diagram Use Case

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
<i>Aktor/actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>
<i>Asosiasi/association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
<i>Ekstensi/extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan
<i><<extend>></i> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
<i>Generalisasi/generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Menggunakan/Include/uses</i> <i><<include>></i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan

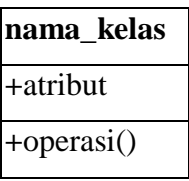



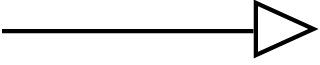
	memerlukan <i>use case</i> ini untuk menjalankan fungsinya
---	--


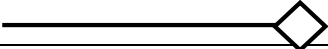
Sumber: (Rosa & Shalahuddin, 2018).

2.4.2. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa & Shalahuddin, 2018). simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2 di bawah ini:

Tabel 2.2 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)





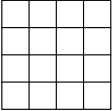
Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)



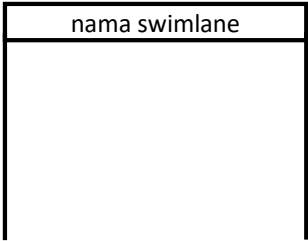
Sumber: (Rosa & Shalahuddin, 2018).

2.4.3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Shalahuddin, 2018). simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2.3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis


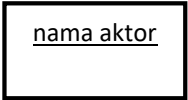
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi




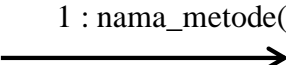
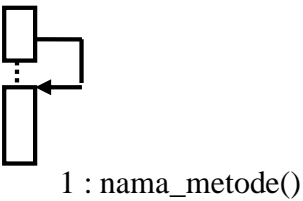

Sumber: (Rosa & Shalahuddin, 2018).

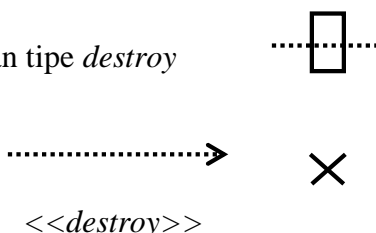
2.4.4. *Sequential Diagram*

Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case*. simbol-simbol yang ada pada *sequence* digram dapat dilihat pada tabel 2.2 di bawah ini :

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  Atau 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor gambar orang, tapi aktor belum tentu

<p>nama aktor tanpa waktu aktif</p>	<p>merupakan orang; biasanya dinyatakan menggunakan kata benda</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>nama objek : nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p> 
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirim data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

<p>Pesan tipe <i>destroy</i></p>  <p>.....→</p> <p><<<i>destroy</i>>></p>	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>
--	---

Sumber : (Rosa & Shalahuddin, 2018)

2.5. Android

Android adalah system operasi untuk telepon seluler berbasis linux sebagai kernelnya. Saat ini android bias disebut raja dari smartphone, hal ini dikarenakan menyediakan platform terbuka (open source) bagi para pengembang untk menciptakan aplikasi mereka sendiri. Awalnya, perusahaan search engine terbesar yaitu Google Inc. membeli Androd Inc. android Inc. didirikan oleh Andi Rubin, Rich Milner, Nick Sears dan Chris White pada tahun 2003, sedangkan pada agustus 2005 Google membeli Android Inc. kemudian untuk mengembangkan android dibentuklah open handset alliance konsorsium dari 34 perusahaan hardware, software dan telekomunikasi Goggle, HTC, Intel, Motorola, Qualqomm, T-Mobile dan Nvidia.

2.6. SQL

Menurut Rosa & Shalahuddin (2018) SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada *Relation DBMS (Database Management System)*.

Menurut Sadeli (2014) MySQL merupakan *Database* yang menghubungkan script php menggunakan perintah *query* dan *escaps character* yang sama dengan php. MySQL mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengizinkan proses yang bisa anda lakukan.

2.7. Pengujian Black – Box

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekeyasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Rosa A.S dan M. Shalahuddin, 2018). Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak benar atau hilang,
- b. Kesalahan *interface*
- c. Kesalahan dalam struktur data atau akses eksternal
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Kelebihan *Black Box* adalah *black box testing* dapat menguji keseluruhan fungsionalitas perangkat lunak. *Black box testing* dapat memilih *subset test* yang secara efektif dan efisien dapat menemukan cacat. Dengan cara ini *black box testing* dapat membantu memaksimalkan testing investment.

Kekurangan *Black Box* adalah ketika *tester* melakukan *black box testing*, tester tidak akan pernah yakin apakah perangkat lunak yang diuji telah benar-benar lolos pengujian.

2.9. Penelitian Terkait

Berikut ini merupakan beberapa Penelitian yang berhubungan dengan laporan skripsi ini yaitu:

Table 2.4 Penelitian Terkait

No	Nama	Judul	Keterangan	Sumber
1	Sardiarinto, Sa'diyah Noor Novita Alfisahrin, Anik Andriani	Rancang Bangun Sistem Reservasi Hotel Menggunakan Metode Waterfall Studi kasus: Hotel Bizz Yogyakarta	Penelitian Ini Menghasilkan Sistem Informasi Reservasi Hotel Menggunakan Pemrograman Delphi. User Yang Merupakan Petugas Front Office Atau Receptionist Di Hotel Dapat Melakukan Transaksi Antara	Jurnal Bianglala Informatika (2015)

			Lain Booking, Checkin, Service Room, Service Restaurant, Checkout, Dan Melihat Dan Mencetak Laporan Sesuai Kebutuhan.	
2	M.Thoha, Miyanto	Analisis Dan Perancangan Sistem Reservasi Hotel D'griya Serang	Menghasilkan menganalisis sistem yang sudah ada tentu mendapat persetujuan dari bagian Electronic Data Prosesing (EDP)	Jurnal PROSISKO (2015)
3	Idria Maita, Arabiatul Adawiyah	Sistem Informasi Reservasi Online Pada Guest House Uin Suska Riau Berbasiskan Web	penelitian ini akan dibangun sebuah Sistem Informasi Guest house dalam mengelola Informasi guest house, reservasi kamar secara online serta fasilitas-fasilitas yang dimiliki oleh guest house. Sistem Informasi guest house ini akan mengolah dan	Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi (2017)

			mengelola semua data yang berhubungan dengan pelanggan atau tamu guest house, ketersediaan kamar, proses check in check out serta payment method atau cara pembayaran.	
4	Ali <i>et al</i>	Silent Monitoring And Cyber Law Awareness Siswa SMK Trisakti Bandar Lampung	Diharapkan siswa SMK Trisakti memahami batasan-batasan dalam menggunakan layanan media sosial ditinjau dari aspek hukum. Selain itu orang tua siswa dilibatkan secara aktif sebagai bagian dari sistem pengawasan anak baik di lingkungan rumah maupun sekolah.	Institut Informatika dan Bisnis Darmajaya (2019)
5	Agarina, Suryadi Karim2 , Sutedi	User-Centered Design	The information displayed at the beginning when the	International Conference on

		<p>Method in the Analysis of User Interface Design of the Department of Informatics System's Website</p>	<p>system is running is adequate, there is already a fairly complete menu that can be accessed by the user. But there are a number of things that need to be improved so that user convenience is even better and userfriendliness characteristics. One of them is that the gallery menu is still not available. Student data also cannot be accessed by students.</p>	<p>Information Technology and Bussiness (ICITB 2019)</p>
--	--	--	--	--