BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Menurut Eko Riswanto (2007:1) dalam bukunya yang berjudul Bahan Ajar Analisa dan Perancangan Sistem, definisi sistem terbagi menjadi dua kelompok pendekatan yaitu menekankan pada prosedurnya dan menekankan pada elemen atau komponennya. Berdasarkan prosedurnya, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Sedangkan definisi sistem yang menekankan pada elemen atau komponennya adalah kumpulan-kumpulan elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Menurut Abdul Kadir (2014:61) bahwa sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sedangkan menurut Sutabri (2012:3) bahwa sistem adalah suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Berbeda dengan penjelasan dari Sutarman (2012:13) bahwa sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

2.2 Informasi

Menurut Eko Riswanto (2007:5), mendefinisikan informasi yaitu data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian *(event)* adalah kejadian yang terjadi pada saat tertentu.

Siklus informasi dijelaskan mulai dari tahap data yang diolah untuk menghasilkan informasi menggunakan model proses yang tertentu. Misalkan suhu dalam fahrenheit diubah ke celcius. Dalam hal ini digunakan model matematik berupa rumus konversi dari derajat fahrenheit menjadi satuan derajat celcius. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima

informasi tersebut, yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterunya yang disebut dengan siklus informasi *(information cycle)*. Siklus ini juga disebut dengan siklus pengolahan data *(data processing cycles).*

2.3 Sistem Informasi

Definisi sistem informasi menurut Eko Riswanto (2007:6) adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan *(building block)* yang terdiri dari:

a. Blok masukan *(input block)*
b. Blok model *(model block)*
c. Blok keluaran *(output block)*
d. Blok teknologi *(technology block)*
e. Blok basis data *(database block)*
f. Blok kendali *(control block)*

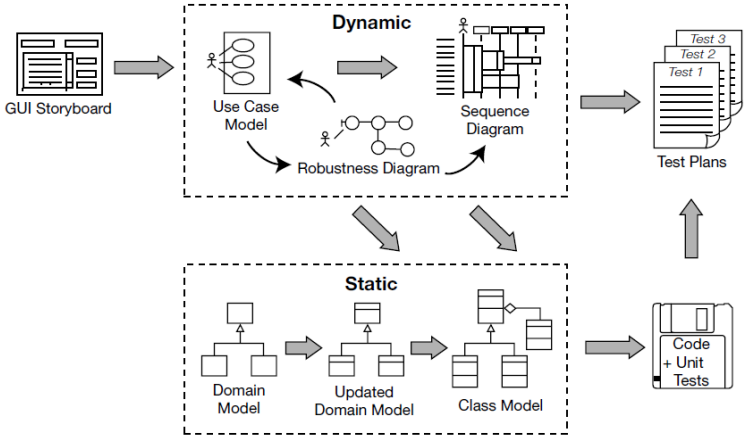
2.4 SKPL

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Dokumen ini akan digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya. Dokumen SKPL dibagi menjadi tiga bagian utama. **Bagian utama** berisi penjelasan tentang dokumen SKPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, referensi dan deskripsi umum. **Bagian kedua** berisi penjelasan secara umum mengenai perangkat lunak yang akan dikembangkan meliputi fungsi dari perangkat lunak, karakteristik pengguna, batasan, dan asumsi yang diambil dalam pengembangan perangkat lunak. **Bagian ketiga** berisi uraian kebutuhan perangkat lunak secara lebih rinci.

2.5 *ICONIX Process*

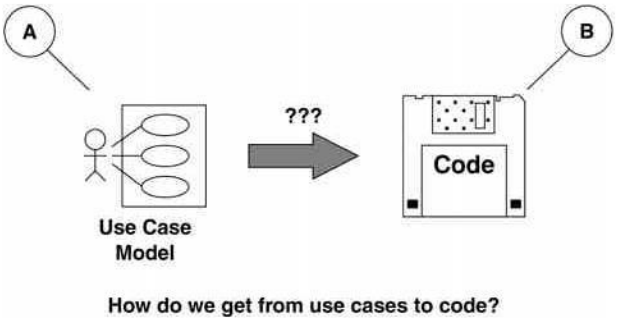
Pendekatan *ICONIX Process* berada ditengah antara pendekatan *Rational Unified Process* (RUP) yang luas dan pendekatan *eXtreme Programming* (XP) yang sangat sempit. Pendekatan *ICONIX Process* didasarkan pada notasi *use case* seperti RUP, tetapi tidak terlalu rumit seperti yang dihasilkan oleh RUP. Lingkup *ICONIX Process* juga sempit dan singkat seperti XP, namun mengutamakan analisis dan desain seperti yang dilakukan XP.

Secara teori, setiap aspek dari UML berpotensi berguna, tetapi dalam prakteknya, tampaknya tidak pernah ada cukup waktu untuk melakukan pemodelan, analisis, dan desain. Selalu ada tekanan dari manajemen untuk beralih ke pembuatan kode, untuk memulai pengkodean lebih awal karena kemajuan pada proyek perangkat lunak cenderung diukur oleh seberapa banyak kode yang ada. *Proses ICONIX*, seperti yang diperlihatkan dalam gambar 3.1, adalah pendekatan minimalis yang disederhanakan yang berfokus pada area yang terletak di antara penggunaan kasus dan kode.



Gambar 2. 1 *ICONIX Process*

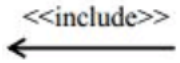
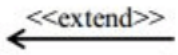
Tujuan utama dari *ICONIX Process* adalah menjembatani bagaimanakah membuat kode program berdasarkan use case yang telah dibuat. Seperti terlihat pada gambar 3.2, Dimana titik A menggambarkan ide atau fungsi apa saja yang harus dilakukan sistem (digambarkan dalam bentuk *use case*). Sedangkan titik B menggambarkan kode - kode program yang komplit, telah diuji, dan telah bisa mengerjakan apa yang dijabarkan pada *use case*.



Gambar 2. 2 Cara Kerja *ICONIX Process*

Berikut akan dijelaskan terkait notasi *use case* yang digunakan dalam *ICONIX Process* yaitu:

Tabel 2. 1 Notasi *Use Case*

No.	Gambar	Keterangan
1		*Aktor*: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case.*
2		*Use case*: Abstraksi dan interaksi antara sistem dan aktor.
3		*Association*: Abstraksi dari penghubung antara aktor dengan *use case*.
4		*Generalisasi*: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan *use case*.
5		*Include*: Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
6		*Extend*: Menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi.

2.6 *Robustness Analysis*

Robustness analysis merupakan diagram informal di dalam UML yang mana bagian dari metode Objectory Jacobson. *Robustness analysis* menunjukkan objek yang berpartisipasi dalam skenario dan bagaimana objek tersebut berinteraksi

satu sama yang lain. Selain itu, *Robustness analysis* juga membantu menjembatani kesenjangan dari *use case* dan domain *class* serta arsitektur software *model-view-control (MVC). Robustness analysis* digambarkan dengan *robustness diagram*.

Tabel 2. 2 Notasi *Robustness Diagram*

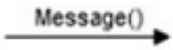
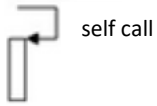
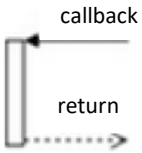
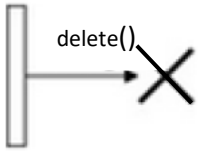
No.	Gambar	Keterangan
1		*Boundary object*: Merepresentasikan antarmuka antara aktor dan sistem.
2		*Controller*: Merepresentasikan logika dari *use case* dan mengoordinasikan kelas-kelas lain.
3		*Entity object*: Mengelola informasi yang dibutuhkan sistem untuk menyediakan fungsionalitas yang diperlukan atau merupakan bentuk representasi data.

2.7 *Sequence Diagram*

Menurut website Dicoding.com, definisi *sequence diagram* adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu, *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

Tujuan utama dari pembuatan *sequence diagram* adalah untuk mengetahui urutan kejadian yang dapat menghasilkan *output* yang diinginkan. Selain itu, tujuan dari *sequence diagram* ini mirip dengan *activity diagram*, seperti menggambarkan alur kerja dari sebuah aktivitas, serta dapat menggambarkan aliran data secara lebih detail, termasuk data atau perilaku yang diterima atau dikirimkan.

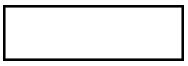
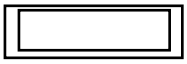
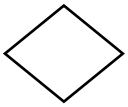
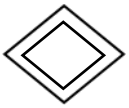
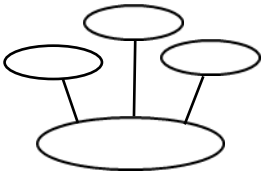
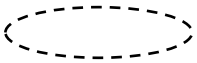
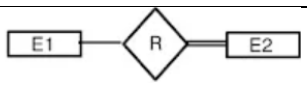
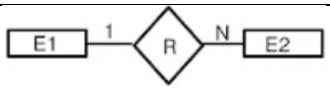
Tabel 2. 3 Notasi *Sequence Diagram*

No.	Gambar	Keterangan
1		*Actor*: Menggambarkan pengguna yang berinteraksi dengan sistem.
2		*Entity class*: Menggambarkan hubungan kegiatan yang dilakukan
3		*Boundary class*: Menggambarkan sebuah form.
4		*Control class*: Menggambarkan boundary dengan tabel.
5		*A Focus of Control and Lifeline*: Menggambarkan tempat mulai dan berakhirnya sebuah *message*.
6		*Aktivasi / Activation*: Menggambarkan ketika suatu objek mengirim atau menerima *message*.
7		*Message*: Menggambarkan objek yang mengirim satu pesan ke objek lain.
8		*Message to Self*: Menggambarkan suatu objek hendak memanggil dirinya sendiri.
9		*Message to Return*: Digambarkan dengan tanda panah yang berbalik ke tempat semula. Komponen ini dijelaskan untuk menunjukkan hasil dari pengiriman *message*.
10		*Destruction*: Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri. Sebaliknya, jika ada *create* maka ada *destroy*.

2.8 *Entity Relationship Diagram (ERD)*

Menurut website Domainesia.com, definisi ERD adalah pemodelan data atau sistem dalam *database* yang sudah sering digunakan oleh banyak lembaga. ERD berfungsi untuk memodelkan struktur dan hubungan antar data yang relatif kompleks. Bentuk ERD seperti diagram yang menjelaskan hubungan antar data. Komponen dalam ERD akan dijelaskan pada tabel dibawah ini:

Tabel 2. 4 Notasi *Entity Relationship Diagram (ERD)*

No.	Gambar	Keterangan
1		*Entity*
2		*Weak Entity*
3		*Relationship*
4		*Identifying Relationship*
5		Atribut
6		Atribut *Primary Key*
7		Atribut *Multivalue*
8		Atribut Komposit
9		Atribut Derivatif
10		*Total Participation of E2 in R*
11		*Cardinality Ratio 1:N For E1:E2 in R*