

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Adapun tinjauan pustaka yang digunakan penulis berisi penelitian terdahulu atau literatur-literatur ilmiah.

2.1.1 Sistem Informasi

Sistem adalah suatu entitas yang terdiri dari bagian-bagian yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu[4]. Beberapa aspek penting merupakan karakteristik sistem. Pertama, interaksi antara komponen sistem sangat penting. Komponen harus bekerja sama dan berbicara satu sama lain untuk mencapai tujuan bersama. Kedua, ruang lingkup operasional sistem ditentukan oleh batasan. Batasan ini menentukan apa yang termasuk dan tidak termasuk dalam sistem. Selain itu, sistem memiliki tujuan yang harus dicapai, yang biasanya berkaitan dengan pemrosesan data atau informasi. Selain itu, sistem juga dapat digunakan untuk mengambil keputusan berdasarkan informasi yang telah mereka proses. Terakhir, sistem dapat menerima input, memprosesnya, dan menghasilkan output yang menguntungkan. Karakteristik ini sangat penting untuk memahami dan menganalisis sistem dalam berbagai situasi, seperti analisis sistem informasi, manajemen, dan rekayasa sistem

2.1.2 Website

Website adalah kumpulan dokumen yang terdiri dari beberapa informasi berbentuk digital dan disusun dalam format *Hyper Text Markup Language* (HTML), yang dapat diakses melalui browser yang terhubung ke jaringan internet melalui *Uniform Resource Locator* (URL)[5]. Ada dua jenis website: yang statis dan yang dinamis. *Website* statis membutuhkan pengubahan konten secara manual, yang berarti membuka *source code*. Sebaliknya, *website* dinamis dapat diubah dengan mudah tanpa membuka *source code* dan selalu diperbarui

2.1.3 Logical Record Structure (LRS)

Logical Record Structure (LRS) adalah struktur data yang digunakan dalam bidang ilmu komputer dan pemrosesan data. Hal ini mengacu pada cara data

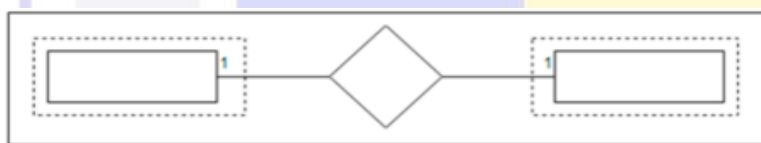
diorganisasikan, direpresentasikan, dan disusun secara logis dalam suatu *file* atau kumpulan data, terlepas dari bagaimana data tersebut disimpan secara fisik pada media penyimpanan.

Ada dua aturan untuk mengkonversi diagram hubungan entitas menjadi *Logical Record Structure* (LRS)[6]. Kedua aturan tersebut adalah:

- Setiap entitas diubah menjadi kotak dengan nama entitas di luar kotak dan atribut di dalam kotak.
- Sebuah hubungan dapat diringkas dalam bidang yang disebut Entitas, atau dipisahkan ke dalam bidang lain

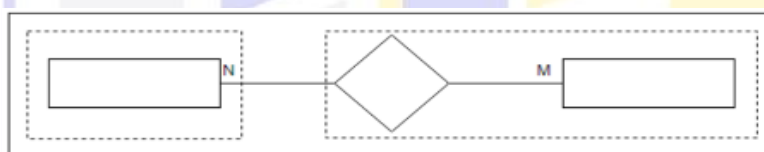
Transformasi dari ERD ke LRS berdasarkan *cardinality*:

- A. Jika hubungan yang terjadi satu ke satu (*One to One*)



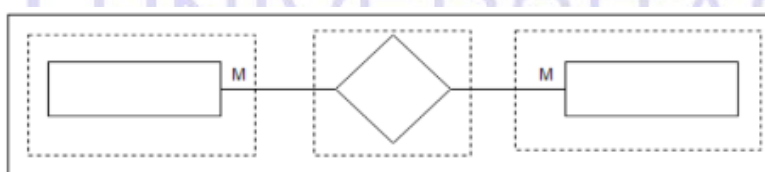
Gambar II.1 *One to One*

- B. Jika hubungan yang terjadi satu ke banyak (*One to Many*)



Gambar II.2 *One to Many*

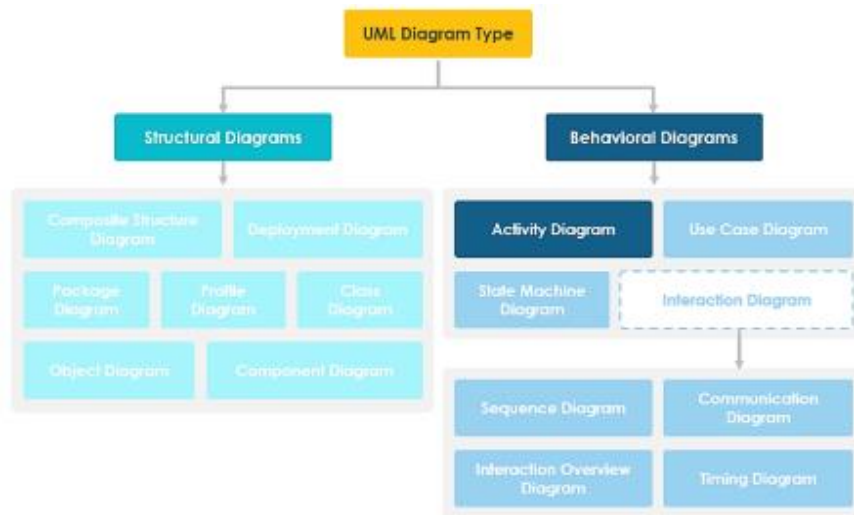
- C. Jika hubungan terjadi banyak ke banyak (*Many to Many*)



Gambar II.3 *Many to Many*

2.1.4 *Unified Modelling Language* (UML)

UML (*Unified Modeling Language*) adalah bahasa berbasis grafik atau gambar untuk memvisualisasikan, menentukan, membuat, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis OOP (*Objcet Oriented*). UML sendiri juga menyediakan standar untuk membuat sistem cetak biru yang mencakup konsep proses bisnis, pembuatan kelas dalam bahasa pemrograman tertentu, skema *database*, dan komponen-komponen yang diperlukan untuk perangkat lunak[7].



Gambar II.4 UML Diagram Type

Adapun tipe-tipe dari diagram UML, antara lain:

1. *Structural Diagram*

a. *Class Diagram*

Adalah diagram yang menggambarkan hubungan antar kelas dalam sistem yang Anda bangun dan bagaimana mereka bekerja sama.

b. *Object Diagram*

Diagram objek menggambarkan objek-objek dalam suatu sistem pada titik waktu tertentu. Diagram objek biasanya disebut sebagai diagram perintah karena penekanannya adalah pada perintah daripada kelas.

c. *Component Diagram*

Merupakan diagram yang menampilkan komponen dalam sistem dan hubungan antara mereka.

d. *Deployment Diagram*

Diagram yang menunjukkan struktur fisik sistem. Menunjukkan perangkat lunak yang berjalan pada perangkat keras yang digunakan untuk mengimplementasikan sistem dan hubungan antar komponen perangkat keras.

2. *Behaviour Diagram*

a. *Activity Diagram*

Diagram aktivitas adalah diagram yang mewakili alur kerja aktivitas berbagai pengguna atau sistem, orang yang melakukan aktivitas, dan urutan aktivitas tersebut.

b. *Use Case Diagram*

Diagram yang menunjukkan peran pengguna dan perannya dalam menggunakan sistem. Dan ini juga menunjukkan interaksi pengguna-sistem dan juga dapat digunakan untuk menjelaskan spesifikasi *use case*.

c. *System Sequence Diagram*

Diagram yang menggambarkan interaksi user dengan sistem secara sekuensial (berurutan).

d. *Collaboration Diagram*

Diagram ini mewakili bentuk lain dari diagram urutan. Diagram ini menunjukkan struktur organisasi suatu sistem di mana pesan dikirim dan diterima.

e. *State Chart Diagram*

Diagram yang menunjukkan bagaimana sistem dapat merespons peristiwa internal atau eksternal. Peristiwa ini menyebabkan perubahan status sistem.

2.1.5 *Entity Relationships Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah model data atau diagram yang menunjukkan hubungan antar entitas atau objek dalam sistem informasi. ERD menunjukkan hubungan antara entitas dan objek dalam sistem, termasuk atribut yang dimiliki oleh setiap entitas dan jenis hubungan yang ada antara mereka. ERD memberikan dasar untuk merancang struktur *database* dan mengimplementasikan sistem dengan lebih efisien dan membantu pengembang sistem memahami lebih baik tentang kompleksitas dan kebutuhan sistem[8].

Berikut langkah-langkah dari pembuatan ERD terdiri atas:

- a. Identifikasi dan penentuan himpunan entitas yang terlibat
- b. Tentukan atribut kunci
- c. Identifikasi dan penentuan relasi antar himpunan entitas
- d. Tentukan derajat/kardinalitas relasi untuk setiap himpunan relasi
- e. Melengkapi himpunan entitas dan relasi dengan atribut bukan kunci.

2.1.6 *Extreme Programming*

Extreme Programming adalah sebuah model yang ditemukan oleh Kent Beck pada tahun 1996. Model ini adalah pengembangan yang sederhana, tidak rumit dan lebih mudah beradaptasi dengan mengawasi persyaratan yang tidak jelas, ambigu, atau

cepat berubah. Model Agile ini sangat menekankan pada kepuasan pengguna. Untuk mengatasi cacat dan kesalahan pada tahap awal, diperlukan laporan berkala dan rilis perangkat lunak yang dilakukan secara bertahap.

Terdapat enam fase di dalam pengembangan pada model ini, yaitu:

1. *Exploration*

Fase pertama pada model ini adalah untuk mengetahui pengembangan arsitektur dan *requirement* produk. Pada tahap ini client memberikan *requirements*, desain dan arsitektur, alat dan aplikasi yang digunakan. Kemudian menentukan jadwal release yang ditentukan oleh tim pengembang dan client. Cara ini dilakukan dengan menggunakan story card yang di tertulis satu sampai dua paragraph non teknis.

2. *Planning*

Tahap perencanaan dapat dimulai tepat setelah tahap eksplorasi selesai dan pada tahap kedua ini yaitu mencari jawaban dari dua pertanyaan dasar yaitu,

- a. Bagian mana yang dapat dibuat dalam tenggat waktu tertentu ?
- b. Apa strategi iterasi selanjutnya ?

Dalam tahap perencanaan disebut dengan “*Planning Game*” yang juga dilakukan dalam dua bagian yaitu *Iteration Planning* dan *Release Planning*.

3. *Release Planning*

Tujuan utama dari release planning adalah untuk menemukan atribut yang dibutuhkan oleh sistem dan rencana pengiriman (*delivery plan*). Perencanaan pelepasan dibagi menjadi tiga tahap yaitu fase eksplorasi, fase komitmen, dan fase pengarahan.

Iteration Planning. Setiap iterasi biasanya dimulai dengan perencanaan iterasi. Pengembang merancang rencana tugas yang mengimplementasikan atribut yang dibutuhkan dari rilis terbaru pada tahap ini. Selama fase ini, pengembang memilih aktivitas yang akan dilakukan dan mengevaluasi biaya, waktu, dan upaya yang dibutuhkan untuk aktivitas yang dipilih. Aktivitas ini dapat dibagi di antara programmer untuk menstabilkan dan menyesuaikan beban kerja.

Iteration to release. Tahap ini mencakup tugas-tugas pengembangan fundamental seperti mendesain, membuat kode, menguji, dan mengintegrasikan. Setiap iterasi dapat memakan waktu 1-4 minggu. Pada iterasi pertama, cerita-cerita tersebut dipilih yang mendesain dan terdiri dari arsitektur produk. Sepasang

pengembang menjalankan aktivitas yang dipilih untuk iterasi terakhir. Pengembang memilih suatu aktivitas dan kemudian membuat desain dasar untuk mengkodekannya. Pengujian fungsional adalah langkah berikutnya setelah penyelesaian pengkodean.

4. Production

Pemrograman ekstrim merilis perangkat lunak dalam jumlah kecil karena merupakan proses tambahan. Sistem rilis berkelanjutan dalam model ini memungkinkan sistem dikembangkan dalam iterasi. Siklus rilis dapat berisi banyak iterasi dan setiap iterasi dapat berlangsung selama satu hingga empat minggu. Fase ini berfokus pada pengiriman perangkat lunak dalam rilis kecil. Selama fase produksi, pengembang menunda kecepatan evolusi sistem untuk menghindari risiko apa pun.

5. Maintenance

Pemeliharaan merupakan proses mendasar untuk setiap kerangka perangkat lunak.

Dalam pemrograman Ekstrim, perangkat lunak diperbarui dan dimodifikasi secara terus-menerus selama jangka waktu tertentu. Fungsionalitas baru dikembangkan sementara yang sebelumnya masih digunakan pada tahap ini. Perubahan yang mungkin menghambat atau menyebabkan masalah dengan produksi dihentikan.

6. Death Phase

Untuk mencapai fase kematian ada dua kemungkinan. Dalam skenario pertama, perangkat lunak dirilis ketika perangkat lunak yang dibangun berisi semua fungsi yang dibutuhkan, kepuasan klien, dan tidak ada cerita. Catatan disimpan dalam bentuk dokumen pendek 5-10 halaman untuk berjaga-jaga jika diperlukan di masa mendatang. Dalam skenario kedua, "kematian entropi" perangkat lunak terjadi dan akan lebih bijaksana untuk menghentikan pengembangan perangkat lunak.

2.2. Penelitian Terkait

Dari hasil studi pustaka terdapat beberapa penelitian yang menjadi referensi penulis untuk mengembangkan penelitian ini, diantaranya yaitu:

A. Penelitian Pertama

1. Judul : Rancang Bangun Sistem Pengaduan Penanganan Kekerasan Seksual Di Universitas Maritim Raja Ali Haji[9]
2. Persamaan
 - User membuat laporan melalui website
3. Perbedaan
 - Admin dengan Satgas dibedakan
 - User dapat mengunggah gambar
4. Hasil Penelitian

Membuat rancangan untuk pembuatan form pengaduan langsung dari website dan akses privilege untuk admin dan satuan petugas adalah sama

B. Penelitian Kedua

1. Judul : HiCare: Aplikasi Pengaduan Kekerasan Seksual berbasis Mobile[10]
2. Persamaan
 - User mendapatkan notifikasi setelah proses diverifikasi
3. Perbedaan
 - Implementasi sistem dibuat untuk android
4. Hasil Penelitian

Membuat rancangan untuk notifikasi laporan jika sudah diproses atau sedang ditindaklanjuti oleh satgas.