

BAB II

LANDASAN TEORI

2.1 Rapat

Menurut Sampebu (2010), rapat merupakan sarana berkumpulnya sekelompok orang untuk menyatukan pikiran, pertimbangan, dan pendapat terhadap suatu urusan, masalah, atau pekerjaan yang diharapkan dapat mencapai suatu mufakat, penyelesaian, dan keputusan. Perencanaan rapat meliputi menentukan topik, menentukan tujuan rapat, menentukan peserta atau pimpinan rapat, menyusun agenda rapat, menyiapkan tempat rapat, dan membagikan notulen atau hasil rapat ke setiap peserta.

2.2 Komunikasi

Komunikasi adalah suatu proses memindahkan informasi dan pengertian (maksud) dari satu orang kepada orang lain (Suprpto, 2006). Komunikasi juga merupakan interaksi antar pribadi yang menggunakan system symbol linguistik, seperti system simbol verbal (kata-kata) maupun nonverbal.

Menurut Rand (2012:12), beberapa alasan pentingnya komunikasi adalah:

1. Komunikasi mendatangkan efektifitas yang lebih besar
2. Komunikasi menempatkan orang-orang pada tempat yang seharusnya.
3. Komunikasi membawa orang-orang untuk terlibat dalam organisasi, meningkatkan motivasi kinerja yang baik, dan meningkatkan komitmen terhadap organisasi.
4. Komunikasi menghasilkan hubungan dan pengertian yang lebih baik antara bawahan, kolega, dan orang-orang di dalam organisasi dan di luar organisasi.

5. Komunikasi menolong orang-orang untuk mengerti perlunya perubahan. Komunikasi meminimalkan permasalahan-permasalahan di dalam keorganisasian seperti konflik, *stress*, demotifasi dan loyalitas.

2.3 Komunikasi dalam Organisasi

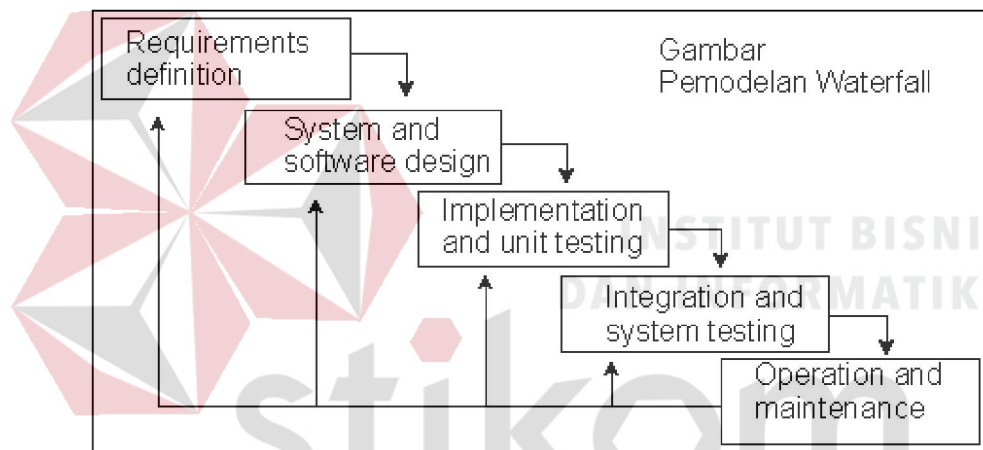
Komunikasi organisasi adalah pengiriman dan penerimaan berbagai pesan organisasi di dalam kelompok *formal* maupun *informal* dari suatu organisasi (Wiryanto, 2006). Berdasar perspektif tradisional (fungsionalis dan objektif), komunikasi organisasi cenderung menekankan pada kegiatan penanganan pesan yang terkandung dalam suatu batas organisasional. Fokusnya adalah menerima, menafsirkan dan bertindak berdasarkan informasi dalam suatu konteks. Tekanannya adalah pada komunikasi sebagai suatu alat yang memungkinkan orang beradaptasi dengan lingkungan mereka. Sedangkan bila dipandang dari perspektif interpretif, komunikasi organisasi adalah proses penciptaan makna atas interaksi yang merupakan organisasi. Komunikasi adalah proses penciptaan makna atas interaksi yang menciptakan, memelihara dan mengubah organisasi (Littlejohn, 2005).

2.4 Node.js

Menurut Rauch (2012:26), *Node.js* merupakan suatu teknologi yang menggunakan *javascript*, tetapi bukanlah di desain untuk *browser* pada *client* melainkan dijalankan pada server. *Node.js* merupakan sebuah *platform* untuk membangun sebuah aplikasi yang membutuhkan jaringan cepat atau *real time*.

2.5 Waterfall Development Model

Model *waterfall* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing / verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement* (Pressman, 2002). Secara umum tahapan pada model *waterfall* dapat dilihat pada gambar berikut :



Gambar 2.1. *Waterfall Methodology*
Sumber: chellessie.blogspot.com/

Berikut tahapan model pengembangan *waterfall*

1. Analisis Kebutuhan

Langkah ini merupakan analisis terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analisis untuk menerjemahkan ke dalam bahasa pemrograman.

2. Desain Sistem

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti *use case diagram*, *class diagram*, dst.

3. Penulisan Kode Program

Penulisan kode program merupakan penerjemahan design dalam bahasa pemrograman, yang dilakukan oleh programmer dalam meterjemahkan transaksi yang diminta oleh pengguna.

4. Penulisan Kode Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

5. Implementasi

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.6 Scrum Development Model

Scrum adalah sebuah kerangka kerja sederhana dimana orang-orang yang akan bekerja dapat menyelesaikan masalah-masalah kompleks dan dapat mengembangkan produk dengan nilai setinggi mungkin secara produktif dan kreatif (Dennis, 2007).

Berikut adalah tahapan dari metode *scrum*

1. Product Backlog

Product Backlog adalah daftar keinginan (*wishlist / desirement*) *Product Owner* untuk produk yang akan dikembangkan oleh Tim Pengembang. *Product Backlog* ini disiapkan dan diurutkan oleh *Product Owner* dan harus transparan bagi semua pihak.

2. Sprint

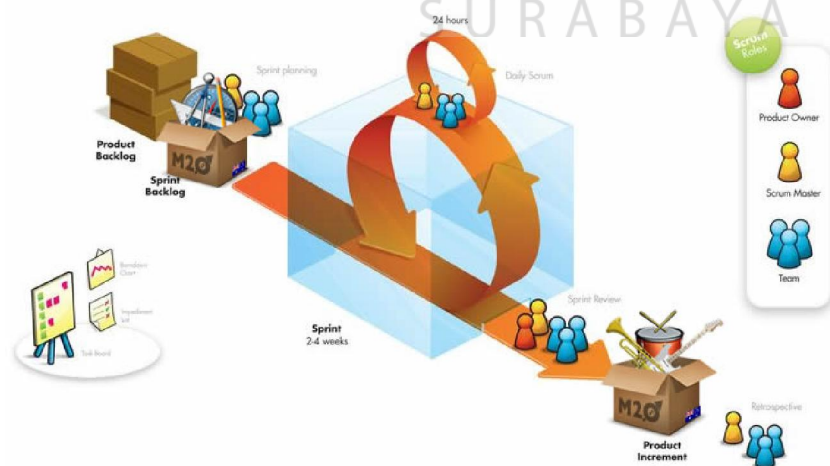
Durasi dari *Sprint* selalu sama/konsisten sepanjang pengembangan produk berlangsung. Artinya apabila Tim *Scrum* telah sepakat untuk memilih durasi.

Di akhir *Sprint*, Tim Pengembang harus menyelesaikan sebuah *backlog* yang digunakan oleh pengguna dan berpotensi untuk dirilis ke lingkungan produksi.

Sprint merupakan pembungkus untuk semua event lainnya dalam *Scrum*.

Event-event lain dalam *Scrum* semuanya dilakukan didalam *Sprint*. *Event-event Scrum* antara lain adalah:

- a. *Sprint Planning*
- b. *Scrum Meeting*



Gambar 2.2. *Scrum Methodology*

Sumber: www.m2o-publishing.com/pro-services/methodology/scrum-methodology/

2.7 Socket io

Menurut Rai (2013:48), *Socket io* merupakan sebuah modul dari *Node.js*. Dibangun diatas *javascipt* untuk membangun aplikasi *real time*. Mempunyai 2 bagian yaitu modul *client* yang berjalan pada *browser*, dan modul *server* untuk *Node.js*. Pada prinsipnya, *Socket io* tidak hanya menggunakan protokol *Websocket*, tetapi juga menggunakan JSONP dan *Ajax Polling*.

Mekanisme kerja socket io adalah sebagai berikut

1. Client, meminta layanan, langkah :

Membuka koneksi client ke server, dengan pertama kali membuat socket dengan perintah `socket()`. Lalu dilanjutkan dengan pengalamatan server, setelah itu komunikasi (mengirim dan menerima data), dapat terjadi dengan menggunakan perintah `write()` dan `read()`

2. Server, menyediakan layanan, langkah:

- a. Melakukan prosedur pembukaan koneksi yang di dalamnya berupa langkah – langkah : membuat socket, mengikat socket, menyiapkan socket menerima koneksi, pengalamatan socket
- b. Looping utama adalah menerima koneksi, dan melakukan komunikasi data (mengirim dan menerima).

2.8 Redis

Redis adalah suatu teknologi penyimpanan data pada *memory* menggunakan metode penyimpanan “*key*” – “*value*” atau lebih sering dikenal dengan nama “*In Memory Database*”.

Redis digolongkan sebagai “*No-SQL*” *database*. NoSQL memiliki empat metode, yaitu:

1. Table-oriented

Metode ini berfungsi untuk mengoptimalkan kinerja penyimpanan data dalam jumlah yang besar, sehingga performa penyajian tetap dapat bagus.

2. Document-oriented

Merupakan database yang berbasis dokumen. Tidak ada tabel, *field* dan *record*, yang ada hanyalah koleksi dan dokumen. Koleksi dapat disamakan dengan tabel dan dokumen disamakan dengan *field*.

3. Graph-oriented database

menggunakan struktur *graph* dengan *node*, *edge* dan *properties* untuk menyimpan datanya.

4. Key-value store

Dalam *key-value store*, data tidak langsung disimpan dalam disk seperti database pada umumnya. Data disimpan dalam memori komputer dan sesekali data dalam memori ditulis ke disk.

2.9 UML

Menurut Nugroho (2005:16), pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antar anggota tim pengembang sistem/perangkat lunak dan antara anggota tim pengembang dengan para pengguna (*end-user*). Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.

2. Penelaahan bagaimana objek-objek dalam sistem saling mengirim pesan (*message*) dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

Beberapa contoh dari UML adalah sebagai berikut

1. Use Case Diagram

Abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara pengguna sebuah sistem.

2. Sequence Diagram

Menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan.

3. State Chart

Memodelkan perilaku dinamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek.

4. Activity Diagram

Menunjukkan suatu alur kegiatan secara berurutan dan mendiskripsikan kegiatan-kegiatan dalam sebuah operasi.

5. Class Diagram

Tujuan utama dari *class* diagram adalah untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna. Diagram kelas biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi.

6. Component Diagram

Diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Pada *component View*, akan difokuskan pada organisasi fisik sistem.

7. Timing Diagram

Digunakan untuk menyelidiki tingkah laku objek sepanjang periode waktu tertentu. Merupakan bentuk khusus dari *sequence* diagram.

8. Object Diagram

Diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu.

9. Deployment Diagram

Merupakan gambarang proses-proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi di dalamnya.

10. Package Diagram

Sekelompok elemen-elemen model yang sama atau dapat berlainan, termasuk paket-paket untuk menciptakan/menggambarkan sifat hirarki.

11. Collaboration Diagram

Diagram yang mengelompokan pesan pada kumpulan diagram sekuen menjadi sebuah diagram..

Dari 11 macam diagram UML tersebut, penulis akan menggunakan 6 macam diagram UML untuk analisis permasalahan di dalam penelitian ini, yaitu: *Use Case* diagram, *Sequence* diagram, *Class* diagram, *Activity* diagram, *Component* diagram, dan *State Chart* diagram.

2.10 Pengujian Perangkat Lunak

Menurut Kaner (2006), Pengujian perangkat lunak merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji.

Terdapat beberapa tipe pengujian perangkat lunak, antara lain:

1. *Installation testing*

Pengujian yang bertujuan untuk menjamin bahwa aplikasi telah ter-*install* dengan benar dan berjalan seperti semestinya, pada sisi pengguna.

2. *Compatibility testing*

Pengujian yang bertujuan untuk mengetahui apakah aplikasi yang baru dapat diintegrasikan dengan aplikasi yang lama, basis data yang telah berjalan, ataupun sistem operasi yang telah beroperasi.

3. *Usability testing*

Pengujian yang berfokus pada *user experience* dari aplikasi yang telah dibuat. Pengujian seringkali dilakukan secara langsung oleh calon pengguna aplikasi.

4. *Acceptance testing*

Pengujian penerimaan oleh pelanggan, seringkali dilakukan pada lingkungan kerja dan oleh mereka sendiri, dikenal sebagai *User Acceptance Test* (UAT).

5. *Functional*

Pengujian yang dilakukan berdasarkan fungsi-fungsi aplikasi yang ada. Pengujian dapat dilakukan oleh pengguna / *customer* dan juga dapat dilakukan pengujian dari internal tim pengembang sendiri.

Penulis akan menggunakan 3 macam pengujian, dari 5 tipe pengujian yang telah disebutkan sebelumnya. Tipe pengujian yang akan digunakan adalah

pengujian *functional* dengan menggunakan tabel pengujian, dan menggabungkan *acceptance test* dan *usability test* dengan cara melakukan pembuatan angket yang akan diisi oleh pengguna aplikasi.

2.11 Skala Likert

Menurut Pershing (2003), Adalah pensklaan berdasarkan distribusi respon subjek terhadap seperangkat pernyataan. Pada skala likert dengan skala lima terdapat lima alternatif jawaban yaitu: sangat setuju, setuju,netral, tidak setuju dan sangat tidak setuju. Pada skala likert lima skala tersebut maka sangat setuju pasti lebih tinggi daripada yang setuju, yang setuju pasti lebih tinggi daripada yang netral, yang netral pasti lebih tinggi daripada yang tidak setuju, sedangkan yang tidak setuju pasti lebih tinggi daripada yang sangat tidak setuju. Namun jarak antara sangat setuju ke setuju dan dari setuju ke netral dan seterusnya tentunya tidak sama, oleh karena itu data yang dihasilkan oleh skala likert adalah data ordinal. Sedangkan cara scoring bahwa sangat setuju 5, setuju 4, netral 3, tidak setuju 2 dan sangat tidak setuju 1 hanya merupakan kode saja untuk mengetahui mana yang lebih tinggi dan mana yang lebih rendah. Perhitungan skala bobot rata-rata setiap pernyataan adalah:

$$M = \frac{\sum X}{\sum N}$$

Keterangan : M = Nilai rata-rata

$\sum X$ = Jumlah nilai

$\sum N$ = Jumlah responden