

## BAB 2 LANDASAN KEPUSTAKAAN

*Customer Relationship Management* (CRM) adalah pendekatan strategi manajemen dalam upaya menciptakan, mengembangkan dan mewujudkan hubungan yang saling menguntungkan dengan pelanggan dalam jangka panjang, khususnya terhadap pelanggan potensial, dalam upaya memaksimalkan *Customer Value* (nilai pelanggan). Aspek-aspek yang terlibat pada *Customer Relationship Management* (CRM) adalah pemasaran, penjualan dan layanan terhadap pelanggan. *Customer Relationship Management* (CRM) juga merupakan strategi inti dalam bisnis yang mengintegrasikan proses-proses dan fungsi-fungsi internal dengan semua jaringan eksternal untuk menciptakan serta mewujudkan nilai bagi para konsumen sasaran secara profit (Buttle, 2004). Dengan pengetahuan yang lebih baik membuat konsumen menjadi lebih pandai dan selektif dalam membeli barang, hal ini menimbulkan sebuah kendala bagi perusahaan dimana setiap perusahaan harus berjuang untuk mempertahankan konsumen yang ada. Saat ini konsumen bisa berpindah ke tempat sejenis yang diinginkan dengan mudah. Hal-hal tersebut membuat perusahaan mulai menganut kembali *Customer Relationship Management* (CRM) untuk meningkatkan penjualan perusahaan.

Berdasarkan penelitian yang telah dilakukan Yuditia Fadhil Anisa dengan judul “*Analisa Perancangan Customer Relationship Management (CRM) Poin Pelanggan Pada Biro Komunikasi Pemasaran PT. Semen Indonesia (Persero) Tbk*”. Dalam penelitian ini peneliti membuat sistem pelayanan aktivitas *Customer Self-Service* terhadap pengelolaan poin pelanggan di Biro Komunikasi Pemasaran. Hasil analisa yang didapat setelah adanya *Customer Self-Service* berbasis web adalah kegiatan poin lebih efisien dalam hal waktu, dimana proses yang lama membutuhkan 1-2 minggu untuk toko mengetahui poin sedangkan proses setelah menerapkan sistem hanya membutuhkan kurang lebih 15 menit. Aspek kepuasan juga dirasakan oleh 10 responden, dimana 48,0% sangat setuju, 45,0% setuju dan 6,7% tidak setuju. Pertanyaan pada aspek ini meliputi *satisfaction* pada sistem seperti sistem telah mempercepat proses poin yang sedang berjalan dan sistem dapat memudahkan toko mengetahui poin yang dimiliki. Untuk aspek pengguna toko sebanyak 33,3% setuju dan 66,7% sangat setuju akan *efficiency* sistem. Hal ini diterima oleh 3 responden toko lebih cepat untuk mengetahui poin yang diperoleh. Hasil pengujian pada sistem yang dirancang telah memenuhi standart penerapan *Customer Relationship Management (CRM)* modul *Sales Force Automation (SFA)* dan *Service Automation (SA)* berdasarkan dasar teori yang ada dengan hasil dapat meningkatkan kepuasan pelanggan dan efisien waktu dari proses kegiatan poin pelanggan.

### 2.1 *Customer Relationship Management (CRM)*

#### 2.1.1 Tujuan *Customer Relationship Management (CRM)*

Adapun tujuan *Customer Relationship Management (CRM)* (Caesar, 2009) yaitu :

1. Menggunakan hubungan dengan pelanggan untuk meningkatkan keuntungan perusahaan
2. Menggunakan informasi yang terintegrasi untuk memberikan pelayanan yang memuaskan

### **2.1.2 Manfaat *Customer Relationship Management* (CRM)**

Manfaat *Customer Relationship Management* (CRM) (Amin, 2008) yaitu :

1. Menjaga pelanggan yang sudah ada
2. Menarik pelanggan baru
3. *Cross selling* yang artinya menjual produk yang dibutuhkan pelanggan berdasarkan pembeliannya
4. *Up selling* yang artinya menawarkan status pelanggan yang lebih tinggi
5. Perusahaan dapat merespon keinginan pelanggan

Pada penelitian ini bertujuan untuk menganalisis pelanggan yang sudah ada dan menarik pelanggan yang baru, serta menerapkan *cross selling* yaitu pada pemesanan *laundry*, dengan kondisi dimana pelanggan dapat menggunakan layanan *laundry* setelah melakukan proses pemesanan atau yang telah mempunyai *ID Customer*, kemudian *up selling* yang selalu menawarkan status pelanggan yang lebih tinggi sehingga dapat mengetahui *promo* atau *event* apapun yang dapat diterima oleh pelanggan dan Everyday Smart Hotel selalu berkeinginan merespon setiap pelanggannya.

### **2.1.3 Sasaran *Customer Relationship Management* (CRM)**

Sasaran utama dari *Customer Relationship Management* (CRM) (Caesar, 2009) antara lain sebagai berikut :

1. Untuk meningkatkan pertumbuhan jangka panjang dan profitabilitas perusahaan melalui pengertian yang lebih baik terhadap kebiasaan pelanggan.
2. Untuk menyediakan umpan balik yang lebih efektif dan integrasi yang lebih baik dengan pengendalian *Return of Investment* (ROI) dalam bisnis tersebut.

### **2.1.4 *Customer Relationship Management* (CRM) Operasional**

*Customer Relationship Management* (CRM) operasional lebih terfokus pada otomatisasi cara perusahaan dalam hubungan dengan pelanggan (Buttle, 2009). Berbagai aplikasi perangkat lunak *Customer Relationship Management* (CRM) memungkinkan fungsi-fungsi pemasaran, penjualan, dan pelayanan dapat berjalan secara otomatis. Beberapa aplikasi terpenting yang dapat diterapkan dalam operasional *Customer Relationship Management* (CRM), yaitu (Machiky, 2011):

- A. Otomatisasi Pemasaran (*Marketing Automation*)
  - 1. Segmen Pasar
  - 2. Manajemen Komunikasi
  - 3. *Even-based Marketing*
- B. Otomatisasi Tenaga Penjualan (*Sales Force Automation*)
  - 1. Proses Penjualan / Manajemen Aktifitas
  - 2. Manajemen Penjualan
  - 3. Manajemen *Contact*
  - 4. *Lead Management*
- C. Otomatisasi Layanan (*Service Automation*)
  - 1. Operasi *contact center* dan *call center*
  - 2. Layanan berbasis *website*
  - 3. Layanan dilapangan

Penelitian ini menerapkan modul *Sales Force Automation* (SFA) pada proses penjualan dan manajemen penjualan yang bertujuan memudahkan pelayanan penjualan atau proses pemesanan di Everyday Smart Hotel, Sedangkan *Service Automation* (SA) pada layanan berbasis *website* dikarenakan sistem yang dibangun berbasis *website* dan layanan dilapangan untuk proses pemesanan secara langsung yang tersedia di *lobby* Everyday Smart Hotel sehingga pelanggan dapat melayani dirinya sendiri.

#### **2.1.4.1 Sales Force Automation (SFA)**

*Sales Force Automation* (SFA) atau otomatisasi tenaga penjualan mengacu pada sistem penjualan dan pemasaran produk yang terintegrasi melalui *web*. Sehingga penjualan produk dapat langsung melaporkan dan memesan produk melalui *web* dengan menggunakan perangkat elektronik seperti komputer. Keuntungan yang dimiliki *Sales Force Automation* (SFA) adalah pelanggan dapat mengetahui aktivitas pemasaran, penjualan dan layanan terhadap pelanggan secara bersama (Payne, 2005). Sasaran yang terdapat pada *Sales Force Automation* (SFA) adalah (Machiky, 2011)

##### A. Proses Penjualan / Manajemen Aktifitas

Proses ini termasuk urutan aktifitas penjualan. Dimana arahan kepada *sales representation* yang terus menerus disetiap tahap yang berlainan dalam proses penjualan.

##### B. Manajemen Penjualan

Merupakan sarana yang mungkin bagi manajer perusahaan dan eksekutif dalam mengakses kebutuhan untuk aktifitas penjualan sebelum, selama,

dan setelah order. Sarana ini memungkinkan *manager* untuk mengatur tim yang berhubungan pada *account* pribadi, *region* dan industri.

#### C. Manajemen *Contact*

Merupakan sarana dengan dengan pengorganisasian dan pengelola data *accros client* milik perusahaan dan yang mengorganisasi prospek

#### D. *Lead Management*

Sarana ini juga dikenal dengan "*Opportunity management and pipeline management*". Bekerja untuk meneliti sejarah *account* pelanggan. Memungkinkan manajemen penjualan untuk secara otomatis menyebarkan petunjuk-petunjuk yang terkait dengan pelanggan di wilayah atau daerah tertentu

### 2.1.4.2 *Service Automation (SA)*

*Service Automation (SA)* atau otomatisasi layanan memungkinkan perusahaan untuk mengotomatisasi fungsi layanan perusahaan terhadap pelanggan melalui *call center, contact person, helpdesk*, layanan berbasis *web* atau bertatap muka langsung dengan pelanggan di lapangan. *Service Automation (SA)* juga menerapkan teknologi informasi internet khususnya *web*, melalui *web* suatu perusahaan dapat memberikan pelayanan sesuai keinginan pelanggan. Manfaat *Service Automation (SA)* bagi perusahaan dan pelanggan yaitu pelayanan kepada pelanggan lebih efektif dan efisien, mengurangi biaya layanan, meningkatkan kualitas layanan, dan meningkatkan produktivitas layanan ke pelanggan.

Adapun macam-macam aktivitas pada modul *Service Automation (SA)* yaitu aktivitas manajemen, agen manajemen, kasus tugas, manajemen kasus, manajemen kontrak, *customer self-service, email respon* manajemen, eskalasi, *inbound* komunikasi manajemen, *outbond* komunikasi manajemen, antrian dan routing, penjadwalan, *scripting*, analisis layanan, layanan manajemen tingkat, manajemen suku cadang, *web* kolaborasi, dan alur kerja teknik (Buttle, 2009). Sedangkan *Customer Self-Service* adalah layanan berbasis dimana pelanggan dapat melayani dirinya sendiri untuk melakukan transaksi, pemesanan, melaporkan pemesanan secara *online* dengan fasilitas *web* yang telah disediakan perusahaan sesuai dengan kebutuhan pelanggan (Payne, 2005).

## 2.2 *Customer Relationship Management (CRM) Evaluation*

*Customer Relationship Management (CRM) Evaluation* merupakan sebuah metode analisis dan evaluasi dari faktor-faktor yang mendukung tingkat keberhasilan suatu sistem *Customer Relationship Management (CRM)*. Seperti pada penelitian sebelumnya, proyek-proyek *Customer Relationship Management (CRM)* yang tingkat keberhasilannya hingga saat ini masih belum memuaskan (Becker et al. 2009; Finnegan dan Currie 2009). Beberapa faktor yang ditemukan mengapa *Customer Relationship Management (CRM)* tidak sesuai harapan yang akan dijelaskan pada Gambar 2.1 merupakan faktor yang dapat menggagalkan

implementasi *Customer Relationship Management* dibedakan menjadi 3 yaitu faktor manusia, proses dan teknologi.



**Gambar 2.1** Faktor Yang Dapat Menggagalkan Implementasi *Customer Relationship Management* (CRM)

Sumber : Friedrich (2011)

Gambar 2.1 menjelaskan bahwa faktor manusia paling berperan yang dapat menggagalkan implementasi *Customer Relationship Management* (CRM) kemudian dilanjutkan oleh beberapa proses beserta teknologi pendukung. Usulan pendekatan yang tepat untuk evaluasi perangkat lunak *Customer Relationship Management* (CRM) adalah melakukan indentifikasi masalah yang dapat menggagalkan suatu sistem *Customer Relationship Management* (CRM). Pendekatan yang tepat adalah mencakup seluruh proses dalam strategi *Customer Relationship Management* (CRM) yang telah didefinisikan sebelum implementasi proyek dimulai. Pendekatan yang diusulkan merupakan evaluasi yang telah dilakukan saat ini atau rekomendasi peningkatan sistem yang dapat digunakan untuk mempermudah proses bisnis (Friedrich, 2011)

### 2.2.1 *Customer Relationship Management* (CRM) Evaluation Approach Criteria

*Customer Relationship Management* (CRM) Evaluation Approach Criteria menjelaskan bahwa kriteria dari segi kualitas, biaya dan fungsional *Customer Relationship Management* (CRM) yang mendukung untuk mengevaluasi sistem *Customer Relationship Management* yang terdapat pada Gambar 2.2 yaitu

| <i>Quality criteria:</i>   | <i>Cost criteria:</i>   | <i>CRM functionality:</i>  |
|--|---|--|
| <ul style="list-style-type: none"> <li>• Portability</li> <li>• Usability</li> <li>• Data Integration</li> <li>• Modifiability &amp; Maintainability</li> <li>• Resources</li> <li>• Training &amp; Support</li> <li>• Reliability &amp; Robustness</li> <li>• Performance &amp; Practicability</li> <li>• Security</li> <li>• Timeliness</li> <li>• Popularity</li> </ul> | <ul style="list-style-type: none"> <li>• System costs (hardware/software licenses)</li> <li>• Preparation and installation costs</li> <li>• Maintenance costs</li> <li>• Resources (consulting, internal)</li> <li>• Training and support</li> <li>• Upgrade costs</li> </ul> | <ul style="list-style-type: none"> <li>• Reporting</li> <li>• Contact Management</li> <li>• Campaign Management</li> <li>• Call Center</li> <li>• Relationship Management</li> <li>• Field Service</li> <li>• Sales Management</li> <li>• Lead/Opportunity Management</li> <li>• Customer Service</li> <li>• Internet</li> <li>• Account Management</li> </ul> |

**Gambar 2.2** Kriteria Evaluasi Sistem *Customer Relationship Management* (CRM)

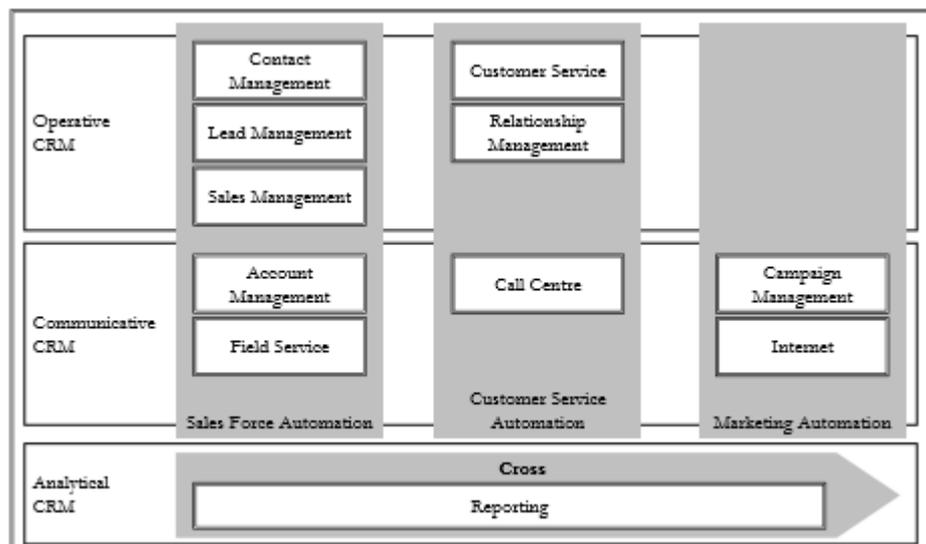
Sumber : Friedrich (2011)

"Kriteria yang berkualitas dapat dinilai dengan melihat serta mengetahui manfaat dan tujuan dari sistem itu." (Farbey et al. 1992). Kriteria evaluasi mempertimbangkan kriteria dari kualitas, biaya dan kebutuhan fungsional. Tujuan dari *quality criteria* adalah sebagai kerangka kerja sistem *Customer Relationship Management* (CRM) yang dapat mencakup *reliability, usability, maintainability, portability, performance (efficiency)* dan *security* yang telah diidentifikasi. Selain itu, mempertimbangkan *data integration, resource, timeliness, popularity, training* dan *support* yang digunakan untuk kebutuhan fungsional sebuah sistem *Customer Relationship Management* (CRM).

Kebutuhan fungsional dapat bervariasi tergantung pada masing-masing perusahaan yang dapat dibagi menjadi 3 aspek, antara lain :

1. *Operative Customer Relationship Management*, yang terdiri dari semua proses dan fungsi mengenai bisnis sehari-hari
2. *Analytical Customer Relationship Management*, yang secara sistematis menganalisis pelanggan dan informasi terkait dengan pelanggan
3. *Communicative Customer Relationship Management*, yang mengontrol dan mendukung hubungan pelanggan (Kemper et al., 2006, p. 172). Dalam beberapa kasus, bidang fungsional dapat ditetapkan *Customer Relationship Management* (CRM) operasional dan komunikatif

Pendekatan ini dengan tujuan untuk melengkapi kebutuhan sistem *Customer Relationship Management* (CRM) sesuai dengan kategori kebutuhan fungsionalnya yang dapat dijelaskan pada Gambar 2.3



**Gambar 2.3 Categories of Evaluation Criteria**

Sumber : Friedrich (2011)

Gambar 2.3 menjelaskan bahwa proses dan fungsi mengenai bisnis sehari-hari yang terdapat pada *operative Customer Relationship Management* (CRM) antara lain *contact management, lead management, sales management, customer*

*service* dan *relationship management*. Kemudian pada kategori *Communicative Customer Relationship Management (CRM)* yang mencakup *account management, field service, call centre, campaign management* dan internet serta *analytical Customer Relationship Management (CRM)* untuk *reporting* dan mendukung hubungan dengan pelanggan.

### **2.2.2 Customer Relationship Management (CRM) Evaluation Approach Method**

Berdasarkan analisis metode yang paling tepat, kriteria dan teknik diidentifikasi untuk mengembangkan pendekatan evaluasi *Customer Relationship Management (CRM)*. Sebelum melakukan evaluasi, perlu mendefinisikan strategi *Customer Relationship Management (CRM)* sebagai dasar untuk evaluasi. Selain itu, sebuah kerangka kerja perusahaan perlu ditetapkan, termasuk mendefinisikan tim proyek, jangka waktu yang telah ditetapkan, tujuan dari proyek serta anggaran proyek yang tersedia untuk setiap tahapnya yang dijelaskan pada Gambar 2.4 yaitu



**Gambar 2.4 Customer Relationship Management Implementation Method**

Sumber : Friedrich (2011)

Pada *Demand Analysis*, menganalisis kerangka kerja dari kebutuhan perusahaan dengan menentukan proses fungsional utama, yang mendasari kebutuhan sistem, termasuk antarmuka yang bergantung pada situasi saat ini serta strategi untuk rekomendasi sistem yang akan diimplementasikan. Semua pemangku kepentingan yang relevan harus terlibat selama fase ini dan ruang lingkup yang perlu didefinisikan sebelumnya.

Pada *Detailed Requirement Specification* merupakan proses yang memiliki target untuk memperoleh kriteria fungsional. Dalam melakukan *Vendor Presentation* bertujuan untuk memperoleh kebutuhan yang lebih dalam pada tingkat cakupan. Vendor wajib untuk memberikan solusi pada permasalahan yang telah dilakukan observasi dan wawancara. Selain itu, kebutuhan fungsional sistem ditemukan untuk memberikan dan menentukan solusi yang spesifik. Setiap pihak yang terlibat mengisi formulir *feedback* untuk menentukan tampilan dan solusi perangkat lunak yang telah diusulkan dan akan dibangun. Selanjutnya, semua materi dapat dianalisis untuk dievaluasi dan diprioritaskan. Terakhir, pada tahap *Decision*, hasilnya dapat diringkas dan didokumentasikan yang akan dipresentasikan kepada pemangku kepentingan yang terlibat. (Friedrich, 2011).

## **2.3 Hotel**

Hotel adalah suatu bentuk bangunan, lambang, perusahaan atau badan usaha akomodasi yang menyediakan pelayanan jasa penginapan, penyedia makanan dan minuman serta fasilitas jasa lainnya yang diperuntukan bagi masyarakat umum, baik mereka yang bermalam di hotel tersebut ataupun mereka yang ingin

menggunakan fasilitas tertentu yang dimiliki hotel itu. Pengertian hotel berdasarkan definisi beberapa para ahli :

1. Pengertian hotel menurut SK Menteri Pariwisata, Pos, dan Telekomunikasi No. KM 37/PW. 340/MPPT-86 adalah “Suatu jenis akomodasi yang mempergunakan sebagian atau seluruh bangunan untuk menyediakan jasa penginapan, makanan dan minuman, serta jasa penunjang lainnya bagi umum yang dikelola secara komersial”
2. Menurut (Sulastiyono, 2013) hotel adalah suatu perusahaan yang dikelola oleh pemiliknya dengan menyediakan pelayanan makanan, minuman dan fasilitas kamar untuk tidur kepada orang-orang yang melakukan perjalanan dan mampu membayar dengan jumlah yang wajar sesuai dengan pelayanan yang diterima tanpa adanya perjanjian khusus.

Berdasarkan definisi para ahli diatas maka penulis menyimpulkan bahwa hotel sebagai suatu usaha jasa yang merupakan sarana pendukung kegiatan pariwisata, dimana pengelolaannya dilakukan secara professional dan didukung oleh tenaga kerja yang memiliki keterampilan baik dalam perhotelan (Maluto, 2013)

## 2.4 Everyday Smart Hotel Malang

Everyday Smart Hotel Malang berdiri pada tanggal 18 November 2013. Pada awalnya, hotel ini dibentuk melalui gagasan kerjasama antara PT. MITRA KARSA yang dimiliki oleh Bapak Teddy dengan PT. JAVA HARBUR TERMINAL yang dikepalai oleh Bapak. Herman. Manajemen Hotel *Goodwood* Asia memperluas investasi dan pengelolaan hotel di Malang, Jawa Timur, yakni Everyday Smart Hotel yang diyakini mampu mengoptimalkan potensi pariwisata. *Goodwood* Asia menawarkan konsep Everyday Smart Hotel sekelas bintang dua atau budget hotel dengan fasilitas terbatas yaitu kamar, ruang makan dan ruang pertemuan saja.



**Gambar 2.5 Logo Everyday Smart Hotel Malang**

Sumber : *Management Office* Everyday Smart Hotel Malang

Everyday Smart Hotel di kota Malang adalah jaringan hotel yang ketiga setelah Kuta (Bali) dan Mangga Besar (Jakarta). Minat investor untuk membangun jaringan hotel *Goodwood* Asia di daerah cukup tinggi bahkan mereka yang hanya punya lokasi tanah strategis saja mampu difasilitasinya dengan pihak perbankan sehingga dapat merealisasikan pendirian *budget* hotel di daerah. Alasan *Goodwood* Asia membangun hotel di kota Malang karena masih banyak destinasi menarik yang belum memiliki infrastruktur termasuk hotel yang memadai. Belum lagi, infrastruktur yang harus dibenahi, mulai dari bandara, pelabuhan, stasiun, dan lain-lain. Wisata bukan lagi hanya sekadar melihat pantai, candi, museum. Kuliner pun telah menjadi tujuan wisata yang digemari. Alasan inilah mengapa

*Goodwood* Asia membuka jaringan hotel di setiap wilayah yang dinilai unggul dari sisi kuliner, salah satunya di Kota Malang.

#### **2.4.1 Visi Everyday Smart Hotel Malang**

Menjadikan Everyday Smart Hotel Malang sebagai hotel pilihan utama di Kota Malang dan sekitarnya.

#### **2.4.2 Misi Everyday Smart Hotel Malang**

Senantiasa bertekad memberikan produk yang bermutu tinggi guna memenuhi harapan pelanggan melalui pengembangan sumber daya dan manajemen kelas dunia.

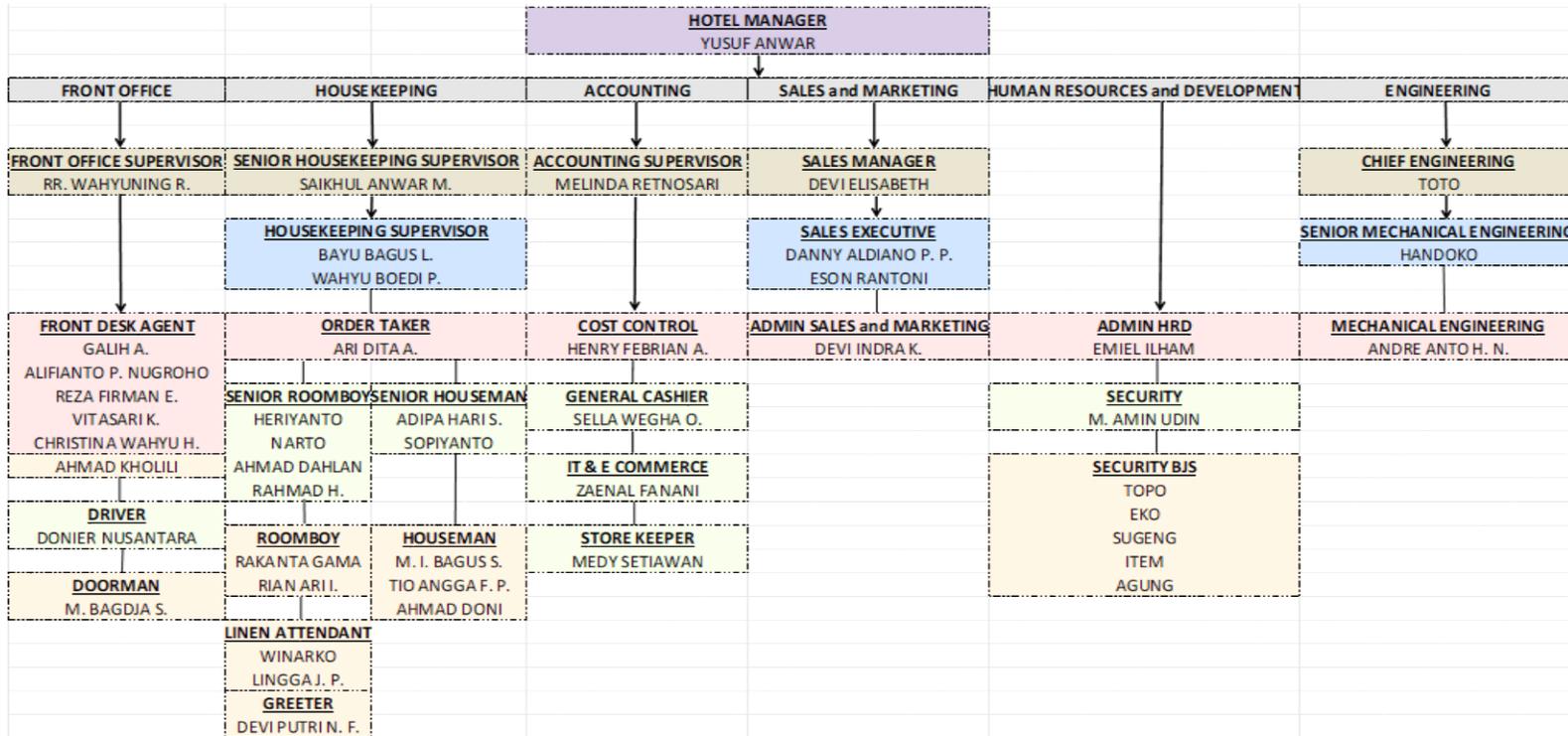
#### **2.4.3 Bisnis dan Strategi Everyday Smart Hotel Malang**

1. *Corporate Marketing*  
Mempersiapkan strategi pemasaran global termasuk laju strategi, segmentasi pasar, rencana biaya pemasaran, dll
2. *Corporate Promotion*  
Mengikuti program promosi dan pameran hotel
3. *E – Commerce*  
Strategi pemasaran melalui internet : Everyday Smart Hotel *Website*, *online travel agent* : traveloka.com, booking.com, sosial media : facebook, instagram, dan lain-lain.
4. *B2B Support*  
Lima *wholesaler* besar: Antavaya, MG Holiday, Mendira, Haryono, KAHA
5. *B2C Activity*  
Penjualan langsung kepada calon konsumen hotel.
6. *Media Exposure*  
Majalah penerbangan, koran, radio, tv, *co-branding* dan lain-lain
7. *Corporate Agreement*  
Kerjasama dengan berbagai perusahaan swasta dan pemerintah
8. *Guest Relationship*  
Menjaga hubungan baik dan komunikasi efektif dengan tamu

#### **2.4.4 Lokasi Everyday Smart Hotel Malang**

Lokasi dari Everyday Smart Hotel bertempat di Jalan Soekarno-Hatta, Malang, Jawa Timur-Indonesia.

### 2.4.5 Struktur Organisasi Everyday Smart Hotel Malang



Gambar 2.6 Struktur Organisasi Everyday Smart Hotel

Gambar 2.6 menjelaskan struktur organisasi di Everyday Smart Hotel Malang pengelolaan saat ini yang dipimpin oleh Bapak Yusuf Anwar dan penelitian ini di dukung oleh bagian *Accounting* pada divisi IT & E-Commerce yaitu Bapak Zaenal Fanani selaku *manager* divisi tersebut.

## 2.5 Proses Bisnis

Proses bisnis merupakan sebuah kumpulan aktivitas yang membutuhkan *input* dan menghasilkan *output* seperti laporan atau peramalan yang bernilai bagi pelanggan (Monk & Wagner, 2013). Menurut (Sparx System, 2004) proses bisnis merupakan kumpulan aktivitas yang dirancang untuk menghasilkan *output* yang spesifik untuk pelanggan atau pasar tertentu. Proses bisnis memiliki komponen yaitu : (1) tujuan yang jelas, (2) adanya masukan, (3) adanya keluaran, (4) menggunakan *resource*. (5) mempunyai sejumlah kegiatan yang ada dibagi dalam beberapa tahapan, (6) dapat mempengaruhi lebih dari satu unit dalam organisasi dan (7) dapat menciptakan nilai atau *value* bagi konsumen.

## 2.6 Business Process Model Notation (BPMN)

*Business Process Modelling Notation* (BPMN) adalah representatif grafis untuk menentukan proses bisnis beserta aktor yang terlibat. *Business Process Modelling Notation* (BPMN) memiliki tujuan untuk menyediakan notasi yang dapat dipahami oleh pengguna bisnis, mulai dari analisis bisnis yang membuat rancangan proses, lalu orang yang menjalankan proses atau yang mengimplementasikan teknologi, serta orang bisnis yang mengelola dan mengontrol proses tersebut (Silver, 2012).

*Business Process Modelling Notation* (BPMN) memiliki empat kategori elemen sebagai penunjang dalam pembuatan *Business Process Modelling Notation* (BPMN). Empat kategori elemen tersebut adalah (Object Management Group, 2011) :

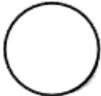
### 2.6.1 Flow Object

*Flow Object* dibagi menjadi 3, yaitu *event*, *activity*, dan *gateway*. Berikut penjelasannya :

#### 1. Event

*Event* merupakan sesuatu yang terjadi dan memiliki dampak dalam proses bisnis. Suatu *event* dapat berasal dari internal dan eksternal suatu proses. *Event* dibagi menjadi tiga yaitu *start event*, *intermediate event*, dan *end event*. Setiap proses selalu memiliki sebuah *start event* untuk menunjukkan awal dari proses bisnis. Tabel 2.1 merupakan penjelasan dari tipe *event*.

**Tabel 2.1 Tipe Event**

| No | Tipe                | Deskripsi   | Simbol  |
|----|---------------------|---|---|
| 1  | <i>Start</i>        | Mendesripsikan dimana suatu proses dimulai.   |  |
| 2  | <i>Intermediate</i> | Mendesripsikan dimana sesuatu terjadi diantara awal dan dan akhir proses. Akan mempengaruhi alur dari proses, tapi tidak akan memulai atau memberhentikan proses. |  |

**Tabel 2.1 Tipe Event (lanjutan)**

| No | Tipe       | Deskripsi                              | Simbol  |
|----|------------|--|---|
| 3  | <i>End</i> | Mendeskrripsikan suatu proses berakhir |  |

Sumber : Object Management Group (2011)

2. *Activity*

*Activity* merupakan tugas yang dilakukan dalam sebuah proses bisnis. *Activity* ditunjukkan dengan kotak dengan ujung bulat dengan nama yang menjelaskan aktivitas yang dilakukan. Terdapat dua macam *activity* yaitu *task* dan *sub process*. Tabel 2.2 Merupakan penjelasan dari tipe *activity*

**Tabel 2.2 Tipe Activity**

| No | Tipe               | Deskripsi   | Simbol   |
|----|--------------------|---|--|
| 1  | <i>Task</i>        | Merupakan aktivitas yang dilakukan pada alur proses.  |   |
| 2  | <i>Sub Process</i> | Merupakan sebuah aktivitas majemuk yang dimasukkan dalam proses. Aktivitas majemuk tersebut dapat dijelaskan dengan lebih <i>detail</i> . |  |

Sumber : Object Management Group (2011)

3. *Gateway*

*Gateway* bertanggung jawab mengontrol bagaimana alur dari sebuah proses bisnis. Tabel 2.3 merupakan penjelasan dari tipe *gateway*

**Tabel 2.3 Tipe Gateway**

| No | Tipe             | Deskripsi   | Simbol  |
|----|------------------|---|---|
| 1  | <i>Exclusive</i> | Sebagai <i>divergence</i> : digunakan untuk membuat jalur alternatif dalam sebuah proses, tapi hanya satu yang dipilih.<br><br>Sebagai <i>convergence</i> : digunakan untuk menggabungkan jalur alternatif. |  |
| 2  | <i>Parallel</i>  | Mendeskrripsikan proses yang dijalankan secara bersamaan.   |  |
| 3  | <i>Inclusive</i> | Mendeskrripsikan sebuah proses yang di pecah menjadi beberapa jalur.  |  |

**Tabel 2.3 Tipe Gateway (lanjutan)**

| No | Tipe           | Deskripsi   | Simbol  |
|----|----------------|---|---|
| 4  | <i>Complex</i> | Mendeskripsikan alur yang kompleks pada sebuah proses bisnis. |  |

Sumber : Object Management Group (2011)

### 2.6.2 Connection

*Connections* adalah elemen yang menghubungkan *flow objects*. Tabel 2.4 merupakan penjelasan dari tipe *connections*.

**Tabel 2.4 Tipe Connections**

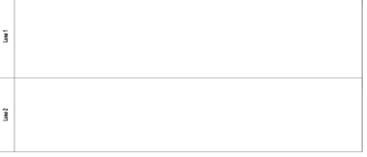
| No | Tipe                 | Deskripsi  | Simbol   |
|----|----------------------|--|--|
| 1  | <i>Sequence Flow</i> | Menunjukkan urutan aktivitas yang dilakukan pada sebuah proses.                                  |   |
| 2  | <i>Association</i>   | Menunjukkan hubungan antara data, teks, artifak lain, dan <i>flow object</i> pada sebuah proses. |   |
| 3  | <i>Message Flow</i>  | Menunjukkan alur pesan antara dua partisipan yang mampu mengirim dan menerima pesan.             |  |

Sumber : Object Management Group (2011)

### 2.6.3 Swimlanes

*Swimlanes* merupakan wadah grafis yang membagi suatu set aktivitas dengan aktivitas lain. Tabel 2.5 merupakan penjelasan tipe *swimlanes*.

**Tabel 2.5 Tipe Swimlanes**

| No | Tipe        | Deskripsi  | Simbol   |
|----|-------------|--|--|
| 1  | <i>Pool</i> | Merupakan wadah yang berisi satu proses dan <i>sequence flow</i> yang menghubungkan aktivitas. |  |
| 2  | <i>Lane</i> | Digunakan untuk merepresentasikan tanggung jawab aktivitas pada sebuah proses.                 |  |

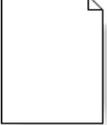
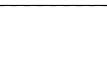
Sumber : Object Management Group (2011)

### 2.6.4 Artifacts

*Artifacts* mempresentasikan sebuah objek diluar sebuah proses. *Artifacts* dapat mempresentasikan data atau catatan yang menjelaskan sebuah proses atau

dapat digunakan untuk mengelola tugas atau proses. Tabel 2.6 merupakan penjelasan tipe *artifacts*.

**Tabel 2.6 Tipe Artifacts**

| No | Tipe               | Deskripsi   | Simbol  |
|----|--------------------|---|---|
| 1  | <i>Data Object</i> | Merepresentasikan informasi yang mengalir pada sebuah proses seperti dokumen bisnis, surat, <i>email</i> , dan lain-lain. |  |
| 2  | <i>Data Store</i>  | Tempat dimana proses dapat membaca atau menulis data.   |  |
| 3  | <i>Annotation</i>  | Menunjukkan informasi tambahan kepada pembaca sebuah diagram <i>Business Process Model Notation</i> (BPMN).               |  |
| 4  | <i>Group</i>       | Memungkinkan untuk mengelompokkan elemen secara informal.   |  |

Sumber : Object Management Group (2011)

## 2.7 Levels of Requirements

Menurut (IBM Corporation Rational Software, 2011) terdapat tingkatan sudut pandang dalam melakukan analisis kebutuhan yang dapat dijelaskan pada Gambar 2.7 yaitu :



**Gambar 2.7 Levels of Requirements**

Sumber : Diadaptasi dari IBM Corporation Rational Software (2011)

### 2.7.1 Business Perspective

*Business Perspective* merupakan sudut pandang berfokus pada fungsi dan proses bisnis yang dibutuhkan untuk memenuhi tujuan dan strategis bisnis. Tujuan *business perspective* adalah memberi pemahaman bagaimana proses bisnis berjalan sehingga didapatkan kebutuhan yang mendukung tujuan bisnis. *Business Perspective* dapat disebut juga sebagai domain masalah.

### 2.7.1.1 Business Process (As-Is vs To-Be)

*Business Process as-is* merupakan proses bisnis yang sedang berjalan disuatu perusahaan atau organisasi. Tujuan dari pemodelan *business process as-is* adalah untuk memahami operasi yang berjalan, siapa saja yang terlibat dalam proses tersebut dan apa hasil dari proses tersebut (ABPMP Internasional, 2013).

Proses analisis dilakukan untuk mengetahui kelemahan dan masalah yang terjadi sehingga dapat ditemukan kesempatan dan solusi untuk perbaikan proses bisnis yang mendatang. Tabel 2.7 merupakan cara dalam melakukan analisis permasalahan.

**Tabel 2.7 Analisis Permasalahan**

|              |  |
|--------------|--|
| Masalah      | [Mendeskripsikan masalah]                            |
| Mempengaruhi | [Pemangku kepentingan yang terpengaruh oleh masalah] |
| Dampak       | [Dampak dari masalah]                                |
| Solusi       | [Solusi beserta manfaatnya]                          |

Sumber : Bittner & Spence (2003)

*Business process to-be* merupakan usulan bagaimana proses bisnis mendatang dapat berjalan dan diimplementasikan berdasarkan analisis *business process as-is*. Tujuannya adalah untuk meningkatkan performa bisnis, meningkatkan efektivitas dan efisiensi proses bisnis, serta memenuhi apa yang diinginkan oleh pemangku kepentingan yang terlibat dalam proses bisnis.

### 2.7.1.2 Kebutuhan Pemangku Kepentingan

Kebutuhan pemangku kepentingan merupakan refleksi dari masalah atau kesempatan dalam bisnis atau pribadi pemangku kepentingan yang perlu ditemukan solusi untuk mendukung tujuan bisnis. Dengan mengetahui kebutuhan pemangku kepentingan, analisis dapat mengerti masalah serta kesempatan yang ada dalam bisnis. Berdasarkan masalah dan kesempatan yang sudah dianalisis, ditemukan solusi yang dapat mendukung dan menyelesaikan masalah pada bisnis (Bittner & Spence, 2003). Langkah-langkah yang dilakukan dalam analisis kebutuhan pemangku kepentingan adalah :

#### 1. Tipe Pemangku Kepentingan

Tipe pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki karakteristik dan hubungan yang sama dengan sistem atau proyek yang menghasilkan sistem. Tipe pemangku kepentingan adalah sebagai berikut :

- a. Pengguna : Tipe pemangku kepentingan yang merupakan pengguna dari sistem. Pengguna merupakan seseorang yang akan memiliki peran aktor dalam *usecase*.

- b. Sponsor : Orang yang melakukan investasi dalam pembangunan suatu sistem. Pemangku kepentingan ini merupakan pengguna tidak langsung dari sistem atau pemangku kepentingan yang terpengaruh pada bisnis melalui sistem.
- c. Pengembang : Merupakan pemangku kepentingan yang terlibat dalam pengembangan sistem.
- d. Otoritas : Ahli dalam bidang tertentu yang berhubungan dengan domain masalah atau solusi seperti ahli teknologi, organisasi pemerintah dan lain-lain.
- e. Pelanggan : Orang atau organisasi yang akan membeli sistem yang sudah siap dipakai

Tabel 2.8 merupakan cara dalam melakukan analisis tipe pemangku kepentingan

**Tabel 2.8 Analisis Tipe Pemangku Kepentingan**

| <b>Tipe Pemangku Kepentingan</b> | <b>Deskripsi</b>                        | <b>Pemangku Kepentingan</b>  |
|----------------------------------|---|--|
| [Tipe pemangku kepentingan]      | [Menjelaskan tipe pemangku kepentingan] | [Menyebutkan pemangku kepentingan yang termasuk dalam tipe pemangku kepentingan] |

Sumber: Diadaptasi dari Bittner & Spence (2003)

## 2. Peran dan Perwakilan Pemangku Kepentingan

Peran pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki peran dan tanggung jawab yang sama pada proyek. Tujuannya adalah untuk mengetahui peran dan tanggung jawab masing-masing pemangku kepentingan sehingga menghindari kesalahan komunikasi dalam pelaksanaan proyek.

Pemangku kepentingan sangat penting dalam proyek karena pemangku kepentingan merupakan sumber dari kebutuhan yang dibutuhkan dalam pembangunan sistem. Dengan mengetahui peran pemangku kepentingan, dapat diambil perwakilan dari pemangku kepentingan tersebut. Tabel 2.9 merupakan cara dalam melakukan analisis peran dan perwakilan pemangku kepentingan.

**Tabel 2.9 Peran dan Perwakilan Pemangku Kepentingan**

| <b>Peran Pemangku Kepentingan</b> | <b>Perwakilan Pemangku Kepentingan</b> | <b>Deskripsi</b>                         |
|-----------------------------------|--|--|
| [Peran pemangku kepentingan]      | [Perwakilan pemangku kepentingan]      | [Menjelaskan peran pemangku kepentingan] |

Sumber: Diadaptasi dari Bittner & Spence (2003)

### 3. Peran Pengguna

Peran pengguna merupakan peran yang dijalankan pengguna saat melakukan interaksi dengan sistem. Peran pengguna dapat merepresentasikan orang atau sistem lain yang dapat berinteraksi dengan sistem.

### 4. Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan pemangku kepentingan merupakan refleksi dari masalah suatu atau kesempatan dalam bisnis atau pribadi pemangku kepentingan yang perlu ditemukan solusi untuk mendukung tujuan bisnis. Identifikasi kebutuhan pemangku kepentingan akan memberikan pandangan mengenai masalah utama dan solusi yang dapat menyelesaikan masalah tersebut (Bittner & Spence, 2003).

Kebutuhan pemangku kepentingan akan dilakukan prioritas untuk memastikan proyek fokus pada bagian paling penting dalam sistem, sehingga tim proyek dapat mengetahui bagian yang perlu dikerjakan terlebih dahulu. Salah satu teknik yang digunakan untuk prioritas kebutuhan adalah aturan MoSCoW. Aturan MoSCoW merupakan aturan yang membagi kebutuhan menjadi empat prioritas yaitu (Stapleton, 1997):

#### a. *Must Have* (Mo)

*Must Have* memiliki arti bahwa suatu fitur dalam proyek harus diselesaikan dan dapat bekerja dengan baik ketika proyek selesai. Fitur tersebut penting karena merupakan solusi dari penyelesaian masalah.

#### b. *Should Have* (S)

*Should Have* memiliki arti bahwa suatu fitur tidak memiliki prioritas utama dalam pengerjaan sistem, tetapi penting dan memiliki nilai bagi pengguna

#### c. *Could Have* (Co)

*Could Have* memiliki arti bahwa suatu fitur yang dapat meningkatkan pengguna sistem, namun jika tidak dapat diimplementasikan tidak berpengaruh pada proyek

#### d. *Won't Have This Time* (W)

*Won't Have This Time* memiliki arti bahwa suatu fitur dapat diimplementasikan setelah pengembangan selesai.

### **2.7.2 User Perspective**

*User Perspective* merupakan sudut pandang pengguna dimana didapatkan solusi bagi pengguna untuk mengatasi masalah yang ada di *business perspective*

### 2.7.2.1 Fitur

Fitur merupakan suatu kemampuan sistem yang menyediakan nilai bagi pengguna serta memenuhi kebutuhan pemangku kepentingan dan pengguna. Suatu fitur dapat memiliki beberapa kebutuhan fungsional (Bittner & Spence, 2003). Fitur akan dilakukan prioritas dengan menggunakan metode MoSCow. Fitur Tabel 2.10 merupakan cara dalam menjelaskan fitur perangkat lunak.

**Tabel 2.10 Fitur Sistem**

| Kode Fitur                             | Fitur          | Deskripsi              | Prioritas         |
|--|----------------|------------------------|-------------------|
| [Kode suatu fitur, sebagai tiap fitur] | [Fitur sistem] | [Deskripsi dari fitur] | [Prioritas fitur] |

Sumber : Bittner & Spence (2003)

### 2.7.2.2 Kebutuhan Fungsional

Kebutuhan fungsional mendeskripsikan apa yang dapat dilakukan oleh sistem. Kebutuhan fungsional memperhatikan masukan serta keluaran yang ada didalam sistem (Bittner & Spence, 2003).

### 2.7.2.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan batasan atau karakteristik yang dimiliki suatu sistem. Kebutuhan non fungsional dapat diimplementasikan dari sisi penggunaan, performa, dukungan dan lain-lain (Bittner & Spence, 2003).

## 2.8 *Unified Modelling Language (UML)*

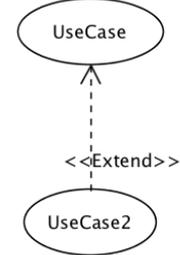
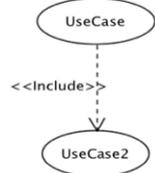
*Unified Modelling Language (UML)* adalah sebuah bahasa standar untuk memodelkan perangkat lunak. *Unified Modelling Language (UML)* dapat digunakan untuk menggambarkan, membangun, dan mendokumentasikan artefak yang diperlukan dalam proses pengembangan sistem (Pressman, 2010). Tujuan dari *Unified Modelling Language (UML)* adalah untuk menggambarkan sebuah mekanisme pemodelan yang mudah dipahami dalam pembangunan suatu sistem. *Unified Modelling Language (UML)* memiliki berbagai jenis diagram diantaranya *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

### 2.8.1 *Use Case Diagram*

*Use case diagram* adalah sebuah diagram untuk menggambarkan tingkah laku sebuah sistem yang akan dibangun beserta fungsionalitas yang ada didalam sistem. *Use case diagram* juga digunakan untuk menggambarkan interaksi antara pengguna dengan sistem (Booch, et al., 2007).

Keuntungan menggunakan *use case diagram* adalah mudah dipahami oleh pihak pemangku kepentingan, menunjukkan tujuan yang ingin di capai melalui serta alat komunikasi kebutuhan yang diinginkan pengguna dan pengembangan untuk memastikan sistem yang dibuat dengan yang pelanggan ingin. Tabel 2.11 merupakan penjelasan dari simbol *use case diagram*.

**Tabel 2.11 Simbol Use Case Diagram**

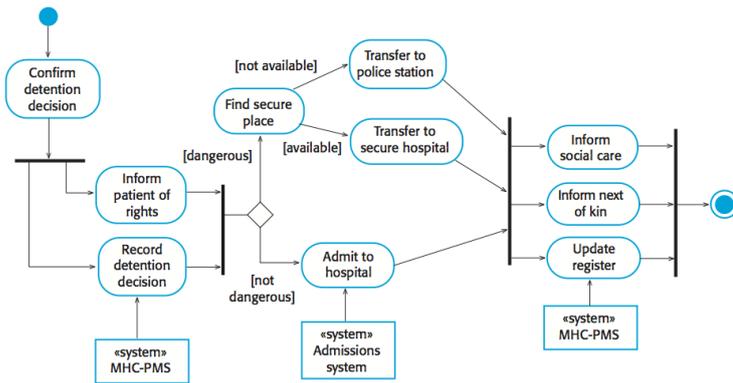
| No | Simbol  | Deskripsi   |
|----|---|---|
| 1  | <p><i>Use Case</i></p>     | <i>Use case</i> merupakan spesifikasi dari kumpulan aksi yang dilakukan oleh sistem.  |
| 2  | <p><i>Actor</i></p>        | Merupakan tipe peran yang dijalankan oleh sebuah entitas untuk berinteraksi dengan sistem. Aktor dapat direpresentasikan oleh manusia, perangkat keras eksternal atau yang lain.  |
| 3  | <p><i>Association</i></p>  | <i>Association</i> merupakan sebuah koneksi antara aktor dan <i>use case</i> . <i>Association</i> mengindikasikan bahwa aktor dapat melakukan suatu <i>use case</i> . Dalam <i>Unified Modelling Language (UML)</i> , <i>association</i> berarti bahwa sebuah aktor terlibat dalam sebuah <i>use case</i> . |
| 4  | <p><i>Extend</i></p>      | <i>Extend</i> merupakan sebuah hubungan dimana memungkinkan kita untuk memodifikasi tingkah laku sebuah <i>use case</i> dasar dengan menambahkan sebuah <i>use case</i> tambahan.   |
| 5  | <p><i>Include</i></p>    | <i>Include</i> menunjukkan bahwa <i>behaviour</i> sebuah <i>included use case</i> termasuk bagian dari sebuah <i>use case</i> dasar. Tujuannya untuk penggunaan kembali sebuah <i>use case</i> yang digunakan berulang kali dalam <i>use case</i> lain.   |

Sumber : Sukamto & Shalahudin (2014)

### 2.8.2 Activity Diagram

*Activity diagram* merupakan sebuah diagram visual yang menggambarkan alur dari sebuah aktivitas seperti sistem, bisnis, alur kerja, atau proses yang lain. *Activity Diagram* berfokus pada aktivitas yang dilakukan dan siapa yang bertanggung jawab pada proses tersebut (Booch, et al., 2007). *Activity Diagram* sesuai dengan Gambar 2.8 digunakan untuk mendefinisikan hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.



**Gambar 2.8 Contoh Activity Diagram**

Sumber : Sommerville (2011)

### 2.8.3 Class Diagram

*Class diagram* digunakan untuk menunjukkan keberadaan dari sebuah *class* dan hubungan antara *class*. Suatu *class diagram* merepresentasikan struktur dari sebuah *class* yang ada di dalam sistem (Booch, et al., 2007). Elemen yang paling penting dalam sebuah *class diagram* adalah *class* dan hubungannya. Tabel 2.12 menjelaskan simbol dari *class diagram*.

**Tabel 2.12 Simbol Class Diagram**

| No | Simbol   | Deskripsi   |
|----|--|---|
| 1  | <p><i>Class</i></p> <pre> classDiagram     class Nama_kelas {         +atribut         +operasi()     }                     </pre> | <i>Class</i> merupakan sebuah <i>template</i> dimana objek dibuat. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> . |
| 2  | <p><i>Association</i></p> <pre> classDiagram     class A     class B     A --- B                     </pre>                        | Relasi yang menunjukkan hubungan antara dua <i>class</i> .  |
| 3  | <p><i>Generalization</i></p> <pre> classDiagram     class A     class B     A &lt; -- B                     </pre>                 | Relasi yang menunjukkan pewarisan dari <i>class</i> utama ( <i>parent</i> ) ke <i>class</i> anak ( <i>child</i> ).                    |
| 4  | <p><i>Dependency</i></p> <pre> classDiagram     class A     class B     A ..&gt; B                     </pre>                      | Relasi yang menunjukkan suatu <i>class</i> bergantung pada <i>class</i> yang lain.  |
| 5  | <p>Agregasi<br/><i>aggregation</i></p> <pre> classDiagram     class A     class B     A o-- B                     </pre>           | Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil.  |

Sumber : Sukanto & Shalahudin (2014)

Dalam pemodelan *class diagram*, terdapat tiga bentuk berbeda berdasarkan tingkat kerinciannya. Tingkat pemodelan *class diagram* melibatkan tiga tahap dalam pengembangan perangkat lunak yaitu definisi kebutuhan, analisis, dan perancangan sistem. Ketiga tahap tersebut antara lain (Kendall, 2004):

1. *Domain - Level Class Diagram*

Pada tahap *domain-level class diagram*, *class diagram* hanya menampilkan nama *class* saja. Tujuan dari *class diagram* pada tahap ini adalah untuk menyediakan istilah umum yang digunakan dalam proyek pengembangan perangkat lunak sehingga para anggota pengembangan dapat memahami dengan jelas proyek yang dikerjakan.

2. *Analysis - Level Class Diagram*

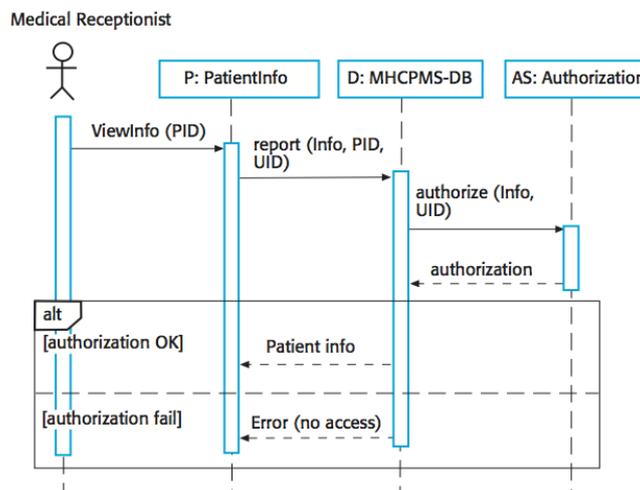
Pada tahap *analysis-level class diagram*, ditambahkan atribut pada tiap *class* di *domain-level class diagram*. *Analysis-level class diagram* tidak perlu menunjukkan operasi pada *class* karena hal tersebut masuk dalam fase analisis.

3. *Design - Level Class Diagram*

Pada tahap *design-level class diagram*, *class diagram* digambarkan secara lengkap operasi dan atribut sesuai dengan standar *class diagram* pada notasi *unified modelling language* (UML).

### 2.8.4 Sequence Diagram

*Sequence diagram* menggambarkan urutan interaksi antara aktor dan objek di dalam sebuah sistem dan interaksi antara objek. Kelebihan dari *sequence diagram* adalah menunjukan urutan fungsi dari sebuah objek ke objek lain. *Sequence Diagram* membantu dalam representasi *detail* dari sebuah *use case* dan digunakan untuk menunjukan interaksi pada suatu *use case* pada suatu skenario dalam perangkat lunak (Sommerville, 2011) Contoh *sequence diagram* dapat dilihat pada Gambar 2.9



Gambar 2.9 Contoh *Sequence Diagram*

Sumber : Sommerville (2011)

## 2.9 Model View Controller (MVC)

*Model View Controller* (MVC) merupakan sebuah pola perancangan dalam perangkat lunak sesuai dengan dimana aplikasi dibagi menjadi tiga bagian yaitu (Basuki, 2014).

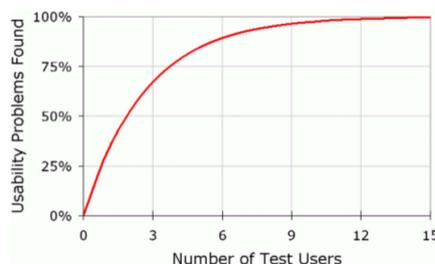
1. *Model*, merepresentasikan data yang digunakan aplikasi. Contoh data misalnya *database*, *Really Simple Syndication* (RSS) atau data yang diperoleh dari pemanggilan *Application Program Interface* (API) dan aksi yang melibatkan operasi *create*, *read*, *update* dan *delete*.
2. *View*, adalah informasi yang ditampilkan kepada *user* melalui *browser*. Biasanya berupa *file Hypertext Markup Language* (HTML) atau kode *Hypertext Preprocessor* (PHP) yang menyusun *template* untuk sebuah *website*.
3. *Controller*, adalah sebuah *business logic* bertugas sebagai jembatan antara *model* dan *view*. *Controller* akan merespon *Hypertext Transfer Protocol* (HTTP) *request* yang datang dari *user* (melalui *browser*), dari *request* ini *controller* akan memanggil *model* untuk mengolah data, kemudian akan memanggil *view* yang sesuai untuk menampilkan data hasil olahan *model* dan akhirnya mengirim kembali halaman hasil olah data tersebut ke *browser*.

## 2.10 Kuesioner

Kuesioner merupakan daftar pertanyaan yang akan diajukan kepada responden. Tujuan dari kuesioner adalah untuk mencari jawaban dari sebuah permasalahan yang sedang diteliti. Kuesioner terdiri dari pertanyaan, pernyataan serta isian yang akan dijawab oleh responden. Jenis pertanyaan yang harus dibuat pada kuesioner haruslah dipersiapkan secara seksama karena hal tersebut akan berpengaruh terhadap jawaban-jawaban yang akan dijawab oleh responden. Jika pertanyaan-pertanyaan yang akan diberikan kepada responden tidak baik maka data yang akan didapat menjadi kurang akurat (Hasibuan, 2007).

## 2.11 Test Users

Pada penelitian tentang *usability problems* yang dilakukan oleh Tom Landauer dan Jakob Nielsen di *usability test* dapat dilihat pada gambar 2.10 dibawah ini.



**Gambar 2.10 Hasil pengujian *usability* dengan jumlah pengguna yang dilakukan Tom Landauer dan Jakob Nielsen**

Sumber : (Nielsen, 2000)

Dari Gambar 2.10 dapat diketahui bahwa kurva akan naik tinggi saat pengujian pada pengguna pertama. Saat memperoleh data dari pengguna pertama, didapatkan data yang hampir telah menyentuh keseluruhan aspek *usability* pada desain tersebut. Kemudian pada pengujian pengguna kedua, didapatkan hasil yang mirip pada pengguna pertama. Akan tetapi, dikarenakan tiap orang mempunyai preferensi yang berbeda-beda, pada pengguna kedua, akan didapatkan data *usability* yang baru yang berbeda dari pengguna pertama. Tetapi data yang didapatkan tidak sebanyak data yang didapatkan dari pengguna pertama karena beberapa data yang didapatkan akan memiliki kesamaan hasil dari pengguna sebelumnya. Pada pengujian pengguna ketiga, keempat, dan seterusnya, didapatkan hasil yang sama pada saat pengujian pengguna sebelum sebelumnya. Data tentang *usability* yang didapatkan dari pengguna ketiga, keempat, dan seterusnya tidak lebih banyak dari pengguna sebelum-sebelumnya dikarenakan ada beberapa data yang akan memiliki kesamaan hasil dari pengguna sebelumnya. Dari pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa semakin banyak pengguna yang dilakukan dalam pengujian, semakin sedikit perbedaan data yang didapatkan. Oleh karena itu pengujian hanya dilakukan dengan menggunakan lima pengguna saja dikarenakan dengan menggunakan lima pengguna, sudah didapatkan 85% dari masalah yang ada pada *usability* dari desain tersebut (Nielsen, 2000).

## 2.12 Skala Pengukuran

Skala merupakan alat untuk mengukur poin kuesioner yang telah dibuat. Terdapat empat macam tipe dari skala pengukuran yaitu nominal, ordinal, interval dan rasio. (Jogiyanto, 2008).

1. Nominal

Nominal dapat dikatakan bernilai klasifikasi. Contohnya laki-laki, perempuan, untuk gender.

2. Ordinal

Ordinal dapat dikatakan bernilai klasifikasi dan ada urutannya (order). Contohnya penilaian kurang, baik, sangat baik.

3. Interval

Interval yaitu bernilai klasifikasi, order, dan berjarak (perbedaan dua nilai berarti). Misalnya pada skala likert.

4. Rasio

Rasio yaitu bernilai klasifikasi, order, berjarak serta origin (mempunyai nilai awal). Misalnya unit waktu sebesar 20 menit yang mempunyai nilai awal 0.

Selanjutnya adalah pemberian nilai-nilai ke tiap variabel yang disebut sebagai metode skala. Terdapat skala *rating* dan skala *ranking* yang ada dalam metode skala. Dalam skala *rating* terdapat beberapa skala tertentu yang dapat digunakan untuk memberikan nilai (*rating*) ke suatu variabel diantaranya adalah :

1. Skala Dikotomi (*Dischotomous Scale*)

Skala jenis ini akan memberikan nilai dikotomi contohnya nilai Ya atau Tidak.

2. Skala Kategori (*Category Scale*)

Skala jenis ini akan memberikan nilai beberapa *item* untuk dipilih.

3. Skala Likert (*Likert Scale*)

Skala ini dapat digunakan untuk mengukur para responden ke dalam lima poin atau tujuh poin skala dengan interval antar skala yang sama

4. Skala Perbedaan Semantik (*Semantic Differential*)

Dalam skala ini, digunakan dua buah nilai ekstrim dari responden dimana responden akan diminta untuk menentukan responnya diantara dua nilai tersebut

5. Skala Numerik (*Numeric Scale*)

Skala ini hampir sama dengan skala perbedaan akan tetapi pada skala ini digantikannya ruang semantik dengan angka-angka numerik

6. Skala Penjumlahan Tetap atau Konstan (*Fixed or Constant Sum Scale*)

Pada skala ini responden akan diminta untuk mendistribusikan nilai responnya ke dalam beberapa *item* yang telah disediakan dengan jumlah yang tetap

7. Skala Stapel (*Stapel Scale*)

Skala ini digunakan tidak hanya untuk mengukur niattas respon dari responden tetapi juga arah responnya

8. Skala Grafik (*Graphic Rating Scale*)

Skala ini menggunakan grafik skala dan responden akan memberikan tanda pada tempat di grafik untuk responnya

Sedangkan untuk skala *ranking* digunakan untuk membandingkan antara dua atau lebih objek untuk memilih mana objek yang lebih baik. Beberapa skala *ranking* diantaranya adalah :

1. *Paired Comparison Scale*

Skala ini digunakan untuk memilih satu dari dua objek secara berpasangan

2. *Force Ranking Scale*

Skala ini digunakan untuk mengurutkan secara langsung dan relatif satu terhadap lainnya

3. *Comparative Scale*

Skala ini akan membandingkan dengan standar yang lainnya.

Pada penelitian ini pengembangan sistem *Customer Relationship Management (CRM) Self-service reservation* menggunakan tipe skala pengukuran yaitu interval dan menggunakan variable skala likert dikarenakan untuk mengetahui ukuran penilaian kepuasan para pengguna dalam menggunakan sistem.

## 2.13 Pengujian

Kebutuhan perangkat lunak dapat dibagi menjadi dua, yaitu : kebutuhan fungsional dan non fungsional. Dimana kebutuhan fungsional berkaitan dengan fungsi sistem dan kebutuhan non fungsional mencakup akan reliabilitas, kapasitas dan performansi sistem.

Pengujian fungsional dapat dilakukan dengan menggunakan metode *whitebox* dan *blackbox* untuk keperluan fungsional sistem apakah sudah dapat berjalan dengan baik atau tidak. Sedangkan untuk pengujian non fungsional ada beberapa yang tidak dapat di uji dengan mudah seperti kapasitas yang tidak cocok, dan rentang waktu yang cukup panjang untuk menjamin bahwa perangkat lunak tidak akan bermasalah. (Gede, 2012)

### 2.13.1 Pengujian *White Box*

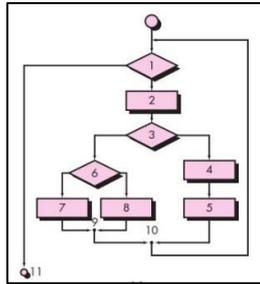
Metode *white box testing* merupakan merupakan teknik pengujian dimana pengujian dilakukan pada kode atau algoritma program. Tujuan dilakukan *white box testing* adalah untuk memastikan tidak ada kesalahan dalam kode atau algoritma program. Kelebihan dari *white box testing* adalah untuk mengetahui kesalahan dalam kode serta memberikan pertimbangan kepada pengembang dalam implementasi perangkat lunak (Agarwal, et al., 2010)

Salah satu metode dari *white box testing* adalah *basis path testing*. *Basis path testing* merupakan pengujian yang memiliki tujuan untuk mengetahui kompleksitas logika (*cyclomatic complexity*) dalam sebuah program (Pressman, 2010). *Cyclomatic complexity* didapatkan berdasarkan *flow graph* suatu kode atau algoritma. *Cycloamtic complexity* digunakan sebagai dasar dalam membuat jalur eksekusi program. Jalur eksekusi program digunakan sebagai dasar dalam membangun kasus uji yang digunakan untuk menjalankan setiap pernyataan di dalam program dalam satu kali pengujian. *Basis path testing* memiliki empat elemen pengujian yaitu (Pressman, 2010) :

#### 1. *Flow Graph*

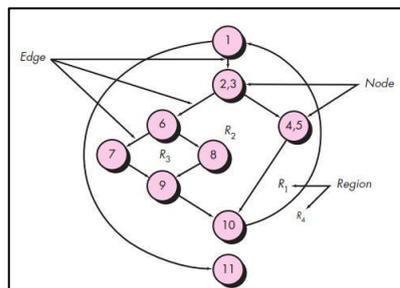
*Flow Graph* mendeskripsikan struktur logika sebuah program. Gambar 2.11 merupakan representasi alur program menggunakan *flow chart*. Gambar 2.12 merupakan representasi alur program menggunakan *flow graph*. Berdasarkan Gambar 2.12 dapat diketahui bahwa bentuk lingkaran merupakan sebuah *node*. Setiap *node* dapat mewakili satu atau lebih prosedural. Setiap *node* yang mewakili suatu kondisi disebut *predicate node*. Tanda panah yang menghubungkan *node* dengan *node* disebut *edge*. *Edge* menggambarkan arah suatu algoritma program

pada *flow chart*. Wilayah yang dibatasi oleh *edge* dan *node* disebut *region*. Ketika menghitung *region*, area diluar *flow graph* dihitung sebagai satu *region*



**Gambar 2.11 Flow Chart Program**

Sumber : Pressman (2012)



**Gambar 2.12 Flow Graph Program**

Sumber : Pressman (2012)

## 2. Cyclomatic Complexity

*Cyclomatic Complexity* merupakan suatu ukuran yang dapat mengetahui kompleksitas logika suatu program. Terdapat tiga cara dalam menghitung *cyclomatic complexity*  $V(G)$  :

- $V(G)$  = jumlah *region* dalam *flow graph*
- $V(G) = E$  (*Edge*) -  $N$  (*Node*) + 2
- $V(G) = P$  (*Predicate Node*) + 1

## 3. Independent Path

*Independent Path* merupakan semua jalur yang dilalui program untuk menghasilkan satu hasil atau kondisi baru. Sebuah *independent path* setidaknya melewati satu *edge* dimana belum dilewati oleh *independent path* sebelumnya

## 4. Test Case

*Test Case* merupakan merupakan data *input* yang digunakan untuk memeriksa alur logika atau kondisi pada masing-masing *independent path*. Berdasarkan (Software Engineering Institute, 1997) nilai *cyclomatic complexity* memiliki arti pada kode, algoritma, atau program. Tabel 2.13 merupakan penjelasan arti nilai *cyclomatic complexity*

**Tabel 2.13 Arti Nilai Cyclomatic Complexity**

| <b>Cyclomatic Complexity</b> | <b>Arti</b>   |
|------------------------------|---|
| 1-10                         | Program mudah dipahami, mudah dilakukan implementasi, mudah dilakukan perbaikan, mudah dilakukan pengujian dan risiko kesalahan program rendah. |
| 11-20                        | Program lebih kompleks, pengujian lebih sulit, dan resiko kesalahan program sedang.   |
| 21-50                        | Program sangat kompleks, pengujian memerlukan usaha karena terdapat banyak jalur eksekusi program, dan resiko kesalahan program tinggi.         |
| Lebih besar dari 50          | Program sulit dilakukan implementasi, perbaikan, pengujian, dan risiko kesalahan program sangat tinggi.   |

### **2.13.2 Pengujian Black Box**

*Black box testing* merupakan teknik pengujian yang hanya melibatkan observasi output dari input nilai tertentu dan tidak ada percobaan untuk analisis kode program. *Black box testing* berfokus pada fungsi sebuah perangkat lunak. *Black box testing* memungkinkan *software engineer* untuk membuat set kondisi input yang dapat dikerjakan oleh semua fungsi dalam perangkat lunak (Agarwal, et al., 2010). Metode pengujian *black box* diperlukan untuk memfokuskan pada keperluan fungsional dari sistem. Kriteria metode *black box testing* berupa *input* dan *scenario* yang dijalankan oleh *user* menghasilkan *output* yang diharapkan. Dalam hal ini perlu mekanisme uji yang berisi *task* atau langkah kerja yang dilakukan pengguna beserta datanya. Jika *output* yang dihasilkan sesuai atau benar maka uji fungsional berhasil, apabila *output* yang dihasilkan tidak sesuai maka uji fungsional gagal (Gede, 2012).

#### **2.13.2.1 Validation Testing**

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kinerja sistem dengan daftar kebutuhan (Indriati, 2010).

#### **2.13.2.2 Usability Testing**

*Usability testing* didefinisikan sebagai metode tingkatan dimana sebuah produk bisa digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektif, efisien, dan memperoleh kepuasan dalam konteks penggunaannya (Lutfiyah, 2012). *Usability testing* adalah tingkatan dari sistem, produk ataupun jasa yang digunakan oleh pengguna untuk mencapai tujuan dengan efektif,

efisiensi dan kepuasan pengguna. Efektivitas merupakan akurasi pengguna dalam meraih tujuan. Efisiensi merupakan akurasi pengguna dalam meraih tujuan yang berhubungan dengan indikator tertentu. Sedangkan kepuasan merupakan perasaan puas terhadap penggunaan dari sebuah produk. Adapun aspek *usability* yaitu (Nielsen, 2012) :

**a. Mudah dipelajari (*learnability*)**

Kualitas sistem yang menunjukkan apakah sistem mudah untuk dipelajari dan digunakan dalam menyelesaikan tugas tertentu.

**b. Efisiensi (*efficiency*)**

Cara yang dapat dilakukan sistem untuk mendukung pengguna dalam melakukan pekerjaannya, memiliki langkah-langkah yang sederhana untuk mendapatkan hasil yang sama.

**c. Mudah diingat (*memorability*)**

Kemampuan sistem untuk mudah diingat, baik dari sisi fitur atau menu-menu yang ada maupun cara pengoperasiannya

**d. Kesalahan dan keamanan (*errors*)**

Perlindungan dan pertolongan kepada pengguna terhadap kondisi dan situasi yang tidak diinginkan dan berbahaya ketika mengoperasikan sistem, misalnya : *menu help* untuk memberi solusi, dan konfirmasi penghapusan berkas.

**e. Kepuasan (*satisfaction*)**

Menunjuk kepada suatu keadaan dimana pengguna merasa puas setelah menggunakan sistem tersebut karena kemudahan yang dimiliki oleh sistem. Semakin pengguna menyukai suatu sistem, secara implisit mereka merasa puas dengan sistem yang dimaksud.

*Usability testing* pada penelitian ini dilakukan dengan dua pengujian yaitu : *full-scale task scenario* dan *Post-Study System Usability Questionnaire* (PSSUQ). *full-scale task scenario* adalah langkah-langkah yang dilihat dari sudut pengguna dalam menggunakan sistem. Sedangkan *Post-Study System Usability Questionnaire* (PSSUQ) adalah untuk mengetahui kepuasan pengguna terhadap aplikasi yang dibangun.

### **2.13.2.3 Full-Scale Task Scenario**

*Full-scale task scenario* mencakup seluruh langkah untuk menyelesaikan *task*. *full-scale task scenario* dapat melaporkan setiap langkah yang dibutuhkan pengguna tertentu atau dapat menjelaskan langkah-langkah yang telah dipersiapkan untuk pengguna baru yang menggunakan sistem. Pada *full-scale task scenario* ini sangat mirip dengan *usecase*, namun skenario ini menyusun langkah-langkah dari sudut pandang pengguna dan bukan dari sudut pandang sistem. Skenario ini menjelaskan bagaimana sistem mendukung skenario yang berorientasi pada pengguna saat awal digunakan. Skenario pada *usability test* akan menunjukkan bagaimana pengguna menyelesaikan sebuah *task* dan menunjukkan apakah antarmuka yang telah dibangun dapat mendukung untuk menyelesaikan skenario. (Usability.gov)

**Appendix: Test Scenario**

|   | Task  | Context & Prerequisites   | UI Flow   | Issues   | Test Scenario   |
|---|---|---|---|--|---|
| 1 | Activate<br>████ toolbar                      | The user is given a general description of what the tool is and that it can be activated from a given site.                                 | <ol style="list-style-type: none"> <li>1. Enter the host site.</li> <li>2. Click on activation button.</li> <li>3. View initial activation screen and click on activation hyperlink.</li> <li>4. Close original window.</li> <li>5. View toolbar being loaded in current window.</li> </ol> | <p>Expectation to and understanding of the initial activation screen.</p> <p>Expectation to and understanding of installation process.</p> | <p>You were told that you can get a tool that can help you █████ both using your favorite sites and links from anywhere, and others. You can get it from your favorite site XXX.com. Just click on this █████ icon.</p> |
| 2 | Browsing and searching<br>(Service provision) | <p>████ toolbar is installed.</p> <p>The user is in a XXX site.</p> <p>The user is asked to search for information and use his familiar</p> | <ol style="list-style-type: none"> <li>1. Enter site XXX.com.</li> <li>2. Search information in usual way.</li> </ol>   | <p>Do they know how to select context?</p> <p>Do they treat it</p>   | <p>Search for information in your regular way. Try to see when you can use</p>  |

**Gambar 2.13 Contoh Full Scale Task Scenario**

(Sumber : Usability.gov)

Pada Gambar 2.13 menjelaskan bagaimana penggunaan *full-scale task scenario*. Kolom pertama *task* merupakan tujuan yang akan dilakukan oleh pengguna untuk menggunakan sistem. Kemudian pada *context* dan *prerequisites* merupakan konteks dan persyaratan yang telah dilakukan oleh pengguna untuk mencakup setiap *task*. *User Interface (UI) flow* merupakan alur di *interaction map*, selanjutnya *issues* merupakan eskpetasi pengujian pada setiap *task* yang dilakukan oleh pengguna. *Test scenario* merupakan pengujian yang dilakukan oleh pengguna beserta tahapan terhadap penggunaan sistem untuk mencapai tujuan pada setiap *task*.

**2.13.2.4 Post-Study System Usability Questionnaire (PSSUQ)**

*Post-Study System Usability Questionnaire (PSSUQ)* merupakan jenis kuesioner yang didesain untuk kepuasan pengguna terhadap aplikasi maupun sistem komputer. Dahulu, *Post-Study System Usability Questionnaire (PSSUQ)* merupakan projek internal milik *International Business Machines (IBM)* dengan nama *System Usability MetricS (SUMS)*. Hingga saat ini, *Post-Study System Usability Questionnaire (PSSUQ)* telah direvisi hingga 3 kali. Pada versi 1 *Post-Study System Usability Questionnaire (PSSUQ)* terdapat 18 item pertanyaan yang telah mewakili 4 karakteristik yaitu (kecepatan dalam menyelesaikan tugas, kemudahan dalam mempelajari sistem, dokumentasi yang berkualitas baik dan informasi secara *online*, serta kecukupan fungsional). Akan tetapi *Post-Study System Usability Questionnaire (PSSUQ)* versi 1 tidak mencakup karakteristik tentang kecepatan akuisisi pada produktivitas. Oleh karena itu dikembangkanlah *Post-Study System Usability Questionnaire (PSSUQ)* versi 2 yang mencakup 19 pertanyaan. Setelah beberapa tahun kemudian, *Post-Study System Usability Questionnaire (PSSUQ)* versi 2 mengindikasikan bahwa ada tiga pertanyaan yang relatif kecil untuk *reliability* oleh karena itu dikembangkanlah *Post-Study System Usability Questionnaire (PSSUQ)* versi 3 yang didalamnya terdapat 16 pertanyaan.

Pertanyaan-pertanyaan tersebut akan mewakili penilaian dari keseluruhan (pertanyaan nomor 1 hingga nomor 16), dan tiga subskala yaitu *System Quality* (SysQual) (pertanyaan nomor 1 hingga nomor 6), *Information Quality* (InfoQual) (pertanyaan nomor 7 hingga nomor 12) dan *Interface Quality* (IntQual) (pertanyaan nomor 13 hingga nomor 15). *Post-Study System Usability Questionnaire* (PSSUQ) versi 3 ini telah diuji reliabilitasnya, dengan mendapatkan nilai reliabilitas pada keseluruhannya yaitu 0.94. Sedangkan untuk reliabilitas tiap-tiap subskalanya adalah SysQual mendapatkan 0.9, InfoQual mendapatkan 0.91, dan IntQual mendapatkan 0.83. (Sauro, 2016). Pada Gambar 2.14 merupakan daftar pertanyaan yang menggunakan *Post-Study System Usability Questionnaire* (PSSUQ) versi 3.

| The Post-Study Usability Questionnaire<br>Version 3 |  | Strongly agree        |                       |                       |                       |                       |                       |                       | Strongly disagree     | NA                    |
|---|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|   |  | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     |                       |                       |
| 1   | Overall, I am satisfied with how easy it is to use this system.  | <input type="radio"/> |
| 2   | It was simple to use this system.  | <input type="radio"/> |
| 3   | I was able to complete the tasks and scenarios quickly using this system.  | <input type="radio"/> |
| 4   | I felt comfortable using this system.  | <input type="radio"/> |
| 5   | It was easy to learn to use this system.   | <input type="radio"/> |
| 6   | I believe I could become productive quickly using this system.   | <input type="radio"/> |
| 7   | The system gave error messages that clearly told me how to fix problems.   | <input type="radio"/> |
| 8   | Whenever I made a mistake using the system, I could recover easily and quickly.  | <input type="radio"/> |
| 9   | The information (such as online help, on-screen messages and other documentation) provided with this system was clear. | <input type="radio"/> |
| 10  | It was easy to find the information I needed.  | <input type="radio"/> |
| 11  | The information was effective in helping me complete the tasks and scenarios.  | <input type="radio"/> |
| 12  | The organization of information on the system screens was clear.   | <input type="radio"/> |
| 13  | The interface* of this system was pleasant.  | <input type="radio"/> |
| 14  | I liked using the interface of this system.  | <input type="radio"/> |
| 15  | This system has all the functions and capabilities I expect it to have.  | <input type="radio"/> |
| 16  | Overall, I am satisfied with this system.  | <input type="radio"/> |

\*The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).

**Gambar 2.14 Post-Study Usability Questionnaire Version 3 Survey**

(Sumber: Chaione.com)

Pertanyaan *Post-Study System Usability Questionnaire* (PSSUQ) menghasilkan empat nilai dari keseluruhan dan memiliki 3 subskala. Aturan untuk menghitungnya yaitu

- Keseluruhan (*overall*) : Rata-rata tanggapan untuk pertanyaan ke 1 hingga pertanyaan ke 16 (semua item)
- Kualitas Sistem (SysQual) : Rata-rata tanggapan untuk pertanyaan 1 hingga pertanyaan ke 6
- Kualitas Informasi (InfoQual) : Rata-rata tanggapan untuk pertanyaan 7 hingga pertanyaan ke 12
- Kualitas Antarmuka (IntQual) : Rata-rata tanggapan untuk pertanyaan 13 hingga pertanyaan ke 15

Skor yang dapat dihasilkan adalah nilai skala antar 1 dan 7, skor yang lebih rendah menunjukkan bahwa tingkat kepuasan yang lebih tinggi. *Post-Study System Usability Questionnaire* (PSSUQ) tidak memerlukan biaya lisensi apapun (Lewis, 2012). Versi awal *Post-Study System Usability Questionnaire* (PSSUQ) menunjukkan skala yang sangat tinggi dan realibilitas subskala. Untuk Versi 3 (Lewis, 2012), reabilitasnya adalah :

- Overall : 0.94
- SysQual : 0.90
- InfoQual : 0.91
- IntQual : 0.83

Seluruh reabilitas yang melebihi 0.80 menunjukkan kehandalan yang cukup berguna sebagai standar pengukuran *usability* (Anastasi, 1976; Landauer, 1997; Nunnally, 1978). Skala dan pertanyaan *Post-Study System Usability Questionnaire* (PSSUQ) berkolerasi di seluruh versi. Gambar 2.15 menunjukkan bahwa contoh terbaik PSSUQ Versi 3 (interval kepuasan mencapai 99%) menggunakan aturan *Post-Study System Usability Questionnaire* (PSSUQ) sehingga skor yang lebih rendah lebih baik daripada nilai yang lebih tinggi.

| Item     | Item Text   | Lower Limit | Mean | Upper Limit |
|----------|---|-------------|------|-------------|
| 1        | Overall, I am satisfied with how easy it is to use this system.   | 2.60        | 2.85 | 3.09        |
| 2        | It was simple to use this system.   | 2.45        | 2.69 | 2.93        |
| 3        | I was able to complete the tasks and scenarios quickly using this system.   | 2.86        | 3.16 | 3.45        |
| 4        | I felt comfortable using this system.   | 2.40        | 2.66 | 2.91        |
| 5        | It was easy to learn to use this system.  | 2.07        | 2.27 | 2.48        |
| 6        | I believe I could become productive quickly using this system.  | 2.54        | 2.86 | 3.17        |
| 7        | The system gave error messages that clearly told me how to fix problems.  | 3.36        | 3.70 | 4.05        |
| 8        | Whenever I made a mistake using the system, I could recover easily and quickly.   | 2.93        | 3.21 | 3.49        |
| 9        | The information (such as online help, on-screen messages, and other documentation) provided with this system was clear. | 2.65        | 2.96 | 3.27        |
| 10       | It was easy to find the information I needed.   | 2.79        | 3.09 | 3.38        |
| 11       | The information was effective in helping me complete the tasks and scenarios.   | 2.46        | 2.74 | 3.01        |
| 12       | The organization of information on the system screens was clear.  | 2.41        | 2.66 | 2.92        |
| 13       | The interface of this system was pleasant.  | 2.06        | 2.28 | 2.49        |
| 14       | I liked using the interface of this system.   | 2.18        | 2.42 | 2.66        |
| 15       | This system has all the functions and capabilities I expect it to have.   | 2.51        | 2.79 | 3.07        |
| 16       | Overall, I am satisfied with this system.   | 2.55        | 2.82 | 3.09        |
| Scale    | Scale Scoring Rule  |             |      |             |
| SysUse   | Average Items 1–6.  | 2.57        | 2.80 | 3.02        |
| InfoQual | Average Items 7–12.   | 2.79        | 3.02 | 3.24        |
| IntQual  | Average Items 13–15.  | 2.28        | 2.49 | 2.71        |
| Overall  | Average Items 1–16.   | 2.62        | 2.82 | 3.02        |

*These data are from 21 studies and 210 participants, analyzed at the participant level.*

**Gambar 2.15 Contoh PSSUQ Version 3 (Interval Kepuasan 99%)**

(Sumber : Quantifying The User Experience 2nd Edition)