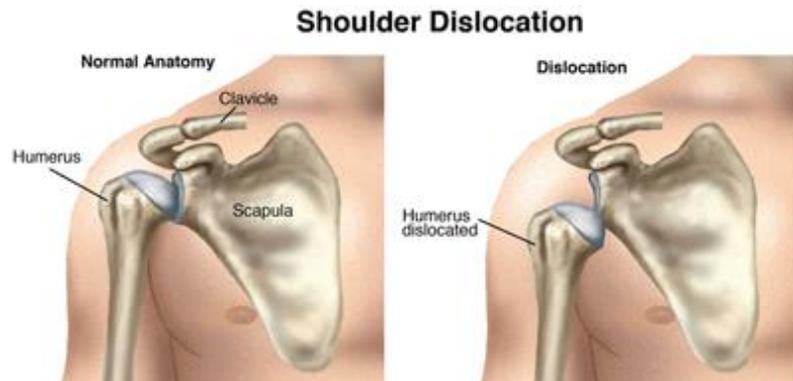


BAB 2

LANDASAN TEORI

2.1 Cedera Bahu dan Terapi Cedera Bahu

Manusia banyak menggantungkan produktifitasnya pada kemampuan lengan sehingga tidak sedikit yang mengalami cedera pada lengan akibat penggunaan yang berlebihan. Salah satu jenis cedera yang terjadi pada mereka yang banyak bekerja berat dengan menggunakan lengan adalah adalah cedera dislokasi bahu [7]. Cedera pada bagian bahu terjadi karena adanya pergeseran tulang di bagian bahu yang bergeser dari soketnya dapat dilihat pada gambar 2.1. Cedera pada bagian bahu ini banyak diakibatkan oleh pergerakan lengan yang mendadak seperti ketika seseorang terjatuh dipermukaan keras dan menumpu pada lengannya, atau ketika lengan seseorang mengalami pukulan atau tendangan yang keras. Akibatnya lengan bagian atas terutama sekitar bahu akan mengalami nyeri yang parah dan berakibat pada sulitnya lengan untuk digerakkan.



Gambar 2. 1 Ilustrasi cedera bahu

Mereka yang mengalami cedera bahu, harus segera memeriksakan dirinya ke dokter, biasanya seorang dokter ahli orthopedi. Dan kemudian pada bagian lengan yang cedera akan dilakukan pemeriksaan dengan menggunakan sinar-X/Rontgen. Yang kemudian, berdasarkan hasil diagnosa dari sinar-X/Rontgen tersebut, akan

menjadi penentu bagi seorang dokter untuk menentukan level cedera bahu yang dialami oleh pasien.



Gambar 2. 2 Contoh hasil pemeriksaan sinar-X/Rontgen pada seorang pasien penderita cedera bahu

Selanjutnya, dokter akan menentukan langkah selanjutnya yang harus dijalani oleh pasien, apakah harus ada tindakan medis lanjut apabila ada kondisi lengan mengalami hal yang lebih parah seperti patah tulang. Atau pasien tersebut dapat melanjutkan tindakan ke tahap terapi oleh seorang fisioterapis.

Dan apabila pasien dapat melanjutkan ke tahap terapi, maka dengan bimbingan seorang fisioterapis, serangkaian kegiatan terapi akan dilakukan untuk mengembalikan kemampuan lengan yang terganggu karena cedera tersebut [1].

Dalam kegiatan terapi, pada umumnya seorang pasien yang mengalami cedera bahu akan melaksanakan terapi dengan melakukan serangkaian gerakan yang telah ditentukan dan dibimbing oleh seorang fisioterapis. Dalam terapi ini, ada dua tahap terapi utama yang dilakukan yaitu tahap pengendoran otot-otot yang kaku akibat cedera. Tahap terapi ini disebut tahap *Loosening*, dimana pada tahap ini serangkaian gerakan dilakkan oleh pasien dengan tidak membawa beban. Kemudian setelah dirasakan cukup, fisioterapi akan membimbing pasien untuk mengikuti tahap berikutnya yang disebut tahap *Strengthening*, yang intinya adalah

tahap penguatan otot-otot lengan. Pada tahap ini, kegiatan terapi yang dilakukan ada yang menggunakan alat bantu dan ada juga yang tidak menggunakan alat bantu.

Beberapa macam gerakan terapi dalam tahapan rehabilitasi yang dilakukan untuk menangani cedera penderita dislokasi lengan adalah sebagai berikut [1]:

1. *Loosening* (Pengendoran Otot)

Adalah tahap dalam fisioterapi yang bertujuan untuk mengendorkan sendi yang kaku. Prinsip dari loosening gerakan dilakukan semampu pasien atau penderita. Beberapa gerakan loosening sebagai berikut:

a. Membuka dan menutup telapak tangan

Pasien dalam keadaan duduk, lalu luruskan tangan yang cedera, kemudian lakukan gerakan menutup dan membuka telapak tangan. Gerakan dilakukan 8-10 pengulangan sebanyak 2 kali.

b. *Wrist Flexion Extension*

Pasien dalam posisi duduk, lalu luruskan tangan yang cedera dan tangan yang satunya menahan bagian tangan yang cedera agar stabil, kemudian gerakan pergelangan tangan ke arah atas dan kebawah. Gerakan dilakukan 10-12 pengulangan sebanyak 2 kali.

c. *Wrist Side Bends*

Pasien dalam posisi duduk, lalu luruskan tangan yang cedera dan tangan yang satunya menahan bagian tangan yang cedera agar stabil, kemudian gerakan pergelangan tangan ke arah samping kanan dan kiri. Gerakan dilakukan 10-12 pengulangan sebanyak 2 kali.

d. *Wrist Circle*

Pasien dalam posisi duduk, lalu luruskan tangan yang cedera dan tangan yang satunya menahan bagian tangan yang cedera agar stabil, kemudian gerakan pergelangan tangan memutar. Gerakan dilakukan 10-12 pengulangan sebanyak 2 kali.

2. *Strengthening* (Penguatan Otot)

Adalah tahap dalam fisioterapi yang bertujuan untuk mengembalikan, bahkan meningkatkan kekuatan sendi sehingga tidak akan terjadinya cedera berulang atau cedera yang sama. Gerakan ini biasanya menggunakan alat bantu seperti elastic

band, karet ban, handuk atau alat yang ada disekitar kita seperti botol yang diisi pasir maupun air. Lakukan gerakan ini tanpa memaksa lengan yang cedera. Beberapa gerakan Strengthening sebagai berikut :

a. *Elastic Band Wrist Extension*

Pasien dalam posisi duduk, lalu letakkan lengan yang cedera di bagian paha, telapak tangan menghadap kebawah sambil memegang elastic band atau karet, lalu tarik ke arah atas hanya menggunakan pergelangan tangan. Gerakan dilakukan 8-10 pengulangan sebanyak 2 kali.

b. *Elastic Band Wrist Curls*

Pasien dalam posisi duduk, lalu letakkan lengan yang cedera di bagian paha, telapak tangan menghadap keatas sambil memegang elastic band atau karet, lalu tarik ke arah atas hanya menggunakan pergelangan tangan. Gerakan dilakukan 8-10 pengulangan sebanyak 2 kali.

c. *Elastic Band Radial Deviation*

Pasien dalam posisi duduk, lalu letakkan lengan yang cedera di bagian paha, telapak tangan menghadap kesamping sambil memegang elastic band atau karet, lalu tarik ke arah atas hanya menggunakan pergelangan tangan. Gerakan dilakukan 8-10 pengulangan sebanyak 2 kali.

d. Memindahkan benda atau bola

Pasien dalam keadaan duduk, lalu letakkan bola ataupun benda diatas meja, lalu gerakkan tangan untuk mengambil bola, kemudian gerakkan tangan yang memegang bola untuk memindahkan bola ke sisi lain meja.

e. Menekuk tangan

Pasien dalam keadaan duduk, lalu luruskan tangan yang cedera, telapak tangan menghadap keatas kemudian lakukan gerakan menekuk meluruskan tangan. Gerakan dilakukan 8-10 pengulangan sebanyak 2 kali. Lakukan hal yang sama setelahnya namun dengan posisi telapak tangan menghadap kebawah.

f. Gerakkan memegang wajah

Pasien dalam posisi duduk, lalu letakkan lengan yang cedera di bagian paha, kemudian gerakkan tangan kearah daerah wajah untuk menyentuh salah satu

bagian wajah seperti hidung, telinga, mata, mulut atau rambut. Gerakan dilakukan 5-8 pengulangan sebanyak 2 kali.

2.2 *Virtual Reality*

Virtual Reality adalah simulasi lingkungan nyata atau imajiner yang dapat dialami secara visual dalam tiga dimensi lebar, tinggi, dan kedalaman dan yang mungkin selain itu memberikan pengalaman interaktif secara visual dalam gerakan real-time penuh dengan suara dan mungkin dengan umpan balik taktil dan bentuk lainnya. *Virtual Reality* adalah sebuah teknologi berbasis komputer untuk mensimulasikan pendengaran visual dan aspek sensorik lainnya lingkungan yang kompleks *Virtual Reality* menggabungkan teknologi 3D yang memberikan ilusi nyata. VR menciptakan simulasi situasi kehidupan [8]. *Virtual Reality* membutuhkan perangkat yang dirancang untuk tujuan tertentu dalam teknologi ini, sehingga mampu menjadikan orang yang merasakan dunia maya terkecoh dan yakin bahwa yang dialaminya adalah nyata. Beberapa perangkat yang digunakan antara lain: *Leap Motion Controller*, *Force Balls/Tracking Balls*, *Controller Wands*, *Voice Recognition*, *Headset*, *Joysticks / Gamepad*, *Data Gloves*, *Treadmills*, *Motion Trackers/Bodysuits*, dan lain sebagainya [9].



Gambar 2. 3 Contoh aplikasi *Virtual Reality* dalam dunia kedokteran

2.3 *Leap Motion Controller (LMC)*

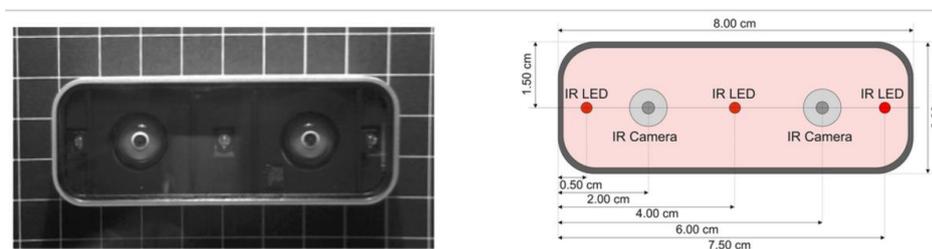
Gesturing adalah bagian alami dari komunikasi manusia dan menjadi semakin penting dalam interaksi AR / VR. *Leap Motion Controller* adalah perangkat baru yang dikembangkan untuk interaksi gerakan oleh *Leap Motion* [10]. Perangkat ini memiliki dimensi kecil 0,5x1.2x3 inci. Untuk menggunakan *Leap Motion Controller*, pengguna perlu menghubungkannya ke komputer dengan USB. Lalu para pengguna letakkan tangan di atas *Leap Motion Controller*.



Gambar 2. 4 Contoh cara menggunakan *Leap Motion Controller*

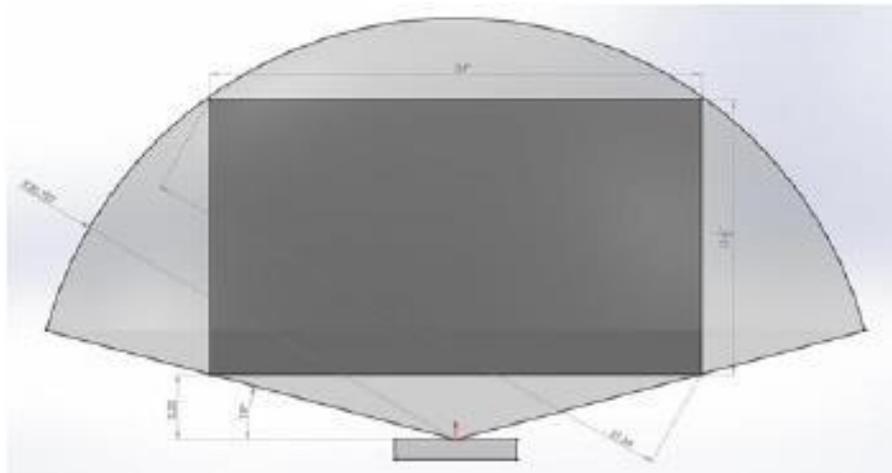
Leap Motion Controller hanya dapat mendeteksi sekumpulan titik yang sudah ditentukan, dan bukan untuk mendeskripsikan bentuk tangan yang sempurna. Alat ini berfungsi untuk mendeteksi gerakan telapak tangan dan jari di atasnya. Data pelacakan yang berisi telapak tangan dan posisi jari, arah, kecepatan dapat diakses menggunakan nya SDK. Menurut [10], *Leap Motion Controller* menyediakan akurasi deteksi sekitar 200 μm .

Gambar 2.2 menunjukkan struktur internal *Leap Motion Controller*. Ada tiga emmitter Cahaya Infra Merah dan dua kamera yang menerima lampu IR.

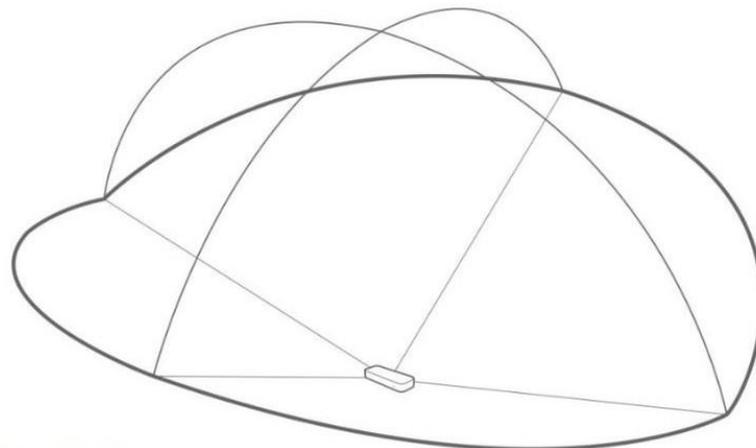


Gambar 2. 5 Struktur internal *Leap Motion Controller*

Leap Motion Controller memperkenalkan sistem pelacakan gerakan dan posisi baru dengan akurasi sub-milimeter. *Leap Motion Controller* sendiri memiliki perangkat keras yang cukup sederhana. Hal utama yang terdapat pada *Leap Motion Controller* yaitu 2 buah kamera dan 3 buah inframerah LED (*Light-Emitting Diode*) [11]. Inframerah ini memiliki panjang gelombang $850 \mu\text{m}$. Jadi pada tahap ini inframerah akan menyebar untuk membentuk sebuah area seperti setengah lingkaran dengan jarak jangkauan maksimal 50 cm. Area jangkauan dari *Leap Motion Controller* ini dapat dilihat pada gambar 2.6 [12] dan dan gambar 2.7 [12]



Gambar 2. 6 Area Jangkauan Inframerah dalam Dua Dimensi



Interaction Area
2 feet above the controller, by 2 feet wide on each side
(150° angle), by 2 feet deep on each side (120° angle)

Gambar 2. 7 Area Jangkauan Inframerah dalam Tiga Dimensi

2.4 Unity 3D

Unity3D adalah *software* yang baik yang merupakan *cross-platform* 3D dan *user-friendly*, mudah bagi pemula dan sangat berguna untuk para ahli. Aplikasi ini sangat membantu bagi siapa saja yang akan membangun sebuah *game* baik itu *desktop*, *mobile*, *website* ataupun *console* [12]. Unity memiliki banyak sekali kegunaan seperti dapat merangkai objek-objek menjadi sebuah lingkungan yang dapat berinteraksi, menambahkan pencahayaan, suara, efek, dan animasi, akan tetapi Unity tidak dapat digunakan untuk mendesain objek atau *modelling*, jika memiliki kebutuhan serupa maka pengguna menggunakan *software* lain untuk melakukan desain seperti Blender dan 3dsmax.



Gambar 2. 8 Tampilan Unity 3D

Terlepas dari itu Unity mendukung banyak sekali *extensi* dan format file lain untuk memperkaya objek dari *game* yang akan dibangun seperti file 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks dan Allegorithmic Zat. *Extensi* ini dapat diimport pada proyek Unity yang kemudian akan diproses oleh grafis Unity. Unity juga telah menyediakan aset sendiri secara gratis maupun berbayar pada Unity Store yang telah tersedia. Di dalam *asset store* terdapat banyak sekali objek, skybox, material, dll secara gratis maupun berbayar. *Scripting* pada Unity secara *default* dibangun dalam mesin *scripting* MonoDevelop. *Scripting* pada Unity dapat dilakukan dengan 3 bahasa yang berbeda seperti C#, JavaScript, dan Boo [14].

2.5 SDK Leap Motion Developer

Sebagai perangkat keras yang inovatif yang terjun pada dunia teknologi maka dibutuhkan dukungan penuh dari pengembang aplikasi ke pengguna. Selain itu juga sebuah teknologi tidak akan berjalan bila tidak ada pengguna yang dapat menggunakannya walaupun teknologi itu sangat bagus.

Untuk itulah Leap Motion, Inc mengeluarkan *Leap SDK* yang dapat digunakan dengan mudah dan gratis. *Leap SDK* memberikan kemudahan-kemudahan dalam mengembangkan aplikasinya, seperti bahasa pemrograman yang disediakan memiliki banyak bahasa diantaranya C, C++, C#, Unity, Objective-C, Java, Python, JavaScript. Juga mendukung beberapa sistem operasi seperti Windows, Linux, dan OSX [9].

2.6 Blender

Blender adalah sebuah perangkat lunak yang berfungsi untuk pemodelan 3 dimensi dan bersifat *open source*. Blender dapat digunakan pada berbagai jenis sistem operasi. Keluaran dari Blender adalah objek-objek 3 dimensi dengan berbagai format, seperti .obj, .3ds, dan lain-lain [16]. Objek keluaran dari Blender dapat digunakan sebagai material dasar pembuatan aplikasi atau permainan dengan menggunakan *game engine* atau perangkat lunak pembuat aplikasi 3 dimensi. Blender memiliki beberapa kegunaan termasuk pemodelan 3D, memberikan tekstur, penyunting gambar bitmap, penulangan, simulasi cairan, simulasi.

2.7 HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web. Website yang dibuat dengan HTML ini, dapat di lihat oleh semua orang yang terkoneksi dengan internet . Tentunya dengan menggunakan aplikasi penjelajah atau browser [17].

Tiap kali kita mengakses dokumen web, maka sesungguhnya kita mengakses dokumen seseorang yang ditulis dengan menggunakan format HTML. Beberapa orang merasa keberatan jika dikatakan HTML adalah sebuah bahasa pemrograman karena struktur yang dimilikinya dianggap terlalu sederhana, kode-kode dibaca oleh

browser baris per baris, dari atas ke bawah. HTML juga tidak memiliki ‘looping’ seperti bahasa pemrograman lain.

Pada HTML dipergunakan hypertext link atau hubungan antara teks dan dokumen lain. Dengan demikian pembaca dokumen bisa melompat dari satu dokumen ke dokumen yang lain dengan mudah.

2.8 PHP

Menurut Betha Sidik (2017), PHP secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *in the fly* yang di eksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemrograman *server side*. Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP [18].

PHP/FI merupakan nama awal dari PHP. PHP – Personal Home Page, FI adalah *Form Interface*. Dibuat pertama kali oleh *Rasmus Lerdoff*. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui *form* yang ditampilkan dalam browser *web*. *Software* ini disebar dan dilisensikan sebagai perangkat lunak *Open Source*. CGI (*Common Gateway Interface*) sendiri menurut Beta Sidik adalah suatu standar yang menghubungkan (*interface*) aplikasi eksternal dengan *server web*.

Kini, PHP adalah kependekan dari PHP:*HyperText Preprocessor*(rekursif mengikut gaya penamaan di *nix), merupakan bahasa utama *script server side* yang disisipkan pada HTML yang dijalankan di *server*.

2.9 JavaScript

Javascript adalah bahasa pemrograman komputer yang dinamis. Ringan dan kebanyakan biasa digunakan sebagai bagian dari halaman web, yang implementasinya memungkinkan sisi klien skrip untuk berinteraksi dengan pengguna dan membuat halaman dinamis. Ini ditafsirkan bahasa pemrograman dengan kemampuan berorientasi objek. JavaScript pertama kali dikenal sebagai

LiveScript, tetapi Netscape mengubah namanya menjadi JavaScript, mungkin karena kegembiraan yang dihasilkan oleh Java. JavaScript membuat penampilan pertamanya di Netscape 2.0 pada tahun 1995 dengan nama LiveScript. Itu inti tujuan umum dari bahasa tersebut telah tertanam di Netscape, Internet Explorer, dan browser web lainnya [18].

2.10 MySQL

MySQL merupakan *database engine* atau *server database* yang mendukung Bahasa database pencarian sql. Mysql adalah sebuah perangkat lunak system manajemen basis data sql atau dbms yang multithread dan multi user. Mysql ab membuat mysql tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL). Mereka juga menjual dibawah lisensi komersial untuk kasu-kasus dimana penggunaannya tidak cocok untuk penggunaan GPL.

Mysql merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu sql (structured query language). Sql adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah dan cepat secara otomatis. Keandalannya suatu system database (DBMS) dapat diketahui dari cara kerja optimizernya dalam melakukan perintah-perintah sql, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, mysql dapat dikatakan lebih unggul dibandingkan database server lainnya dalam hal query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query mysql bisa sepuluh kali lebih cepat dari postgresql dan lima kali lebih cepat dibandingkan interbase [19].

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat open source (tidak berbayar). MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL

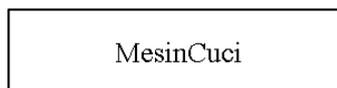
lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP.

2.11 Pemodelan dan UML

UML (*Unified Modelling Language*) merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, pencarian *interface* dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen [20].

2.12 Class Diagram

Class, dalam notasi UML digambarkan dengan kotak. Nama *class* menggunakan huruf besar di awal kalimatnya. Dan diletakkan di atas kotak. Bila *class* mempunyai nama terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar [20].

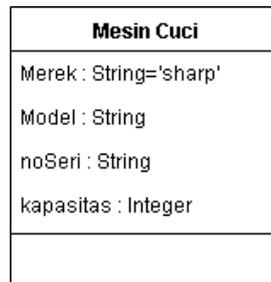


(Sumber : Pemodelan Visual dengan UML[20])

Gambar 2. 9 Notasi Class

1. *Attribute*

Attribute adalah properti dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada obyek dari *class*. Sebuah *class* mungkin mempunyai nol atau lebih *attribute*. Secara konvensi, jika nama *attribute* terdiri atas satu suku kata, maka ditulis dengan huruf kecil. Akan tetapi jika nama *attribute* mengandung lebih dari satu suku kata maka semua suku kata digabungkan dengan suku kata pertama menggunakan huruf kecil dan awal suku kata berikutnya menggunakan huruf besar [20].

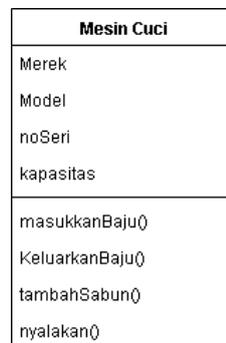


(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 10 Attribute

2. Operation

Operation adalah suatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat dilakukan untuk sebuah *class*. Seperti halnya *attribute*, nama *operation* juga menggunakan huruf kecil semua jika terdiri dari satu suku kata. Akan tetapi jika lebih dari satu suku kata, maka semua suku kata digabungkan dengan suku kata pertama huruf kecil dan huruf awal tiap suku berikutnya dengan huruf besar [20].



(Sumber : Pemodelan Visual dengan UML[20])

Gambar 2. 11 Operation

Berikut adalah tabel operator yang dipakai pada suatu class diagram:

Tabel 2. 1 Operator

Operator	Visibility	Keterangan
+	<i>Public</i>	Sebuah <i>operation</i> atau <i>attribute</i> bisa di akses oleh siapapun
-	<i>Private</i>	Fitur yang hanya digunakan oleh <i>instance</i> dari <i>class</i>
#	<i>Protected</i>	Seperti private tetapi boleh di akses oleh anak-anaknya
-	<i>Package</i>	Bisa diakses langsung oleh <i>instance</i> dengan <i>package</i> sama

2.13 Use Case Diagram

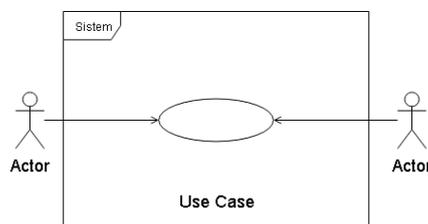
Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna [21].

Dalam *use case*, pengguna biasanya disebut dengan aktor, aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *usecase* adalah bagian dari mode *requirement*. Termasuk disini adalah *problem domain object model* dan penjelasan tentang *user interface*. *Usecase* memberikan spesifikasi fungsi-fungsi yang di tawarkan oleh sistem dari perspektif user.

1. Notasi Use Case

Diagram *use case* menunjukkan 3 aspek dari sistem yaitu *actor*, *use case*, dan sistem/sub sistem boundari. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*. Gambar 2.7 mengilustrasikan *actor*, *use case* dan boundari[20].



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 12 Use Case Model

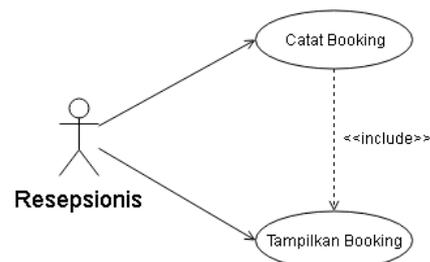
2. Identifikasi Actor

Dalam mengidentifikasi *actor*, terlebih dahulu menentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target

sistem. *Actor* adalah *abstraction* dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case* [20].

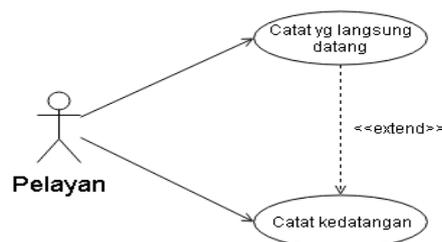
3. Stereotype

Stereotype adalah sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan stereotype digunakan simbol “<<” di awalnya dan ditutup ”>>” di akhirnya. <<extend>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu yang dipenuhi. Sedangkan <<include>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. Biasanya <<include>> digunakan untuk menghindari pengkopian suatu *use case* karena sering dipakai [20].



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 13 Use Case Inclusion



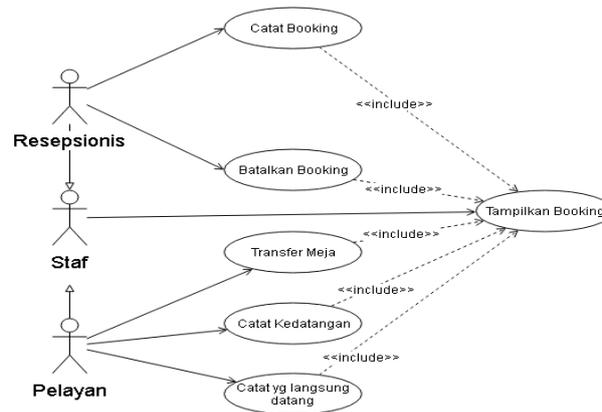
(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 14 Use Case Extension

4. Level Use Case

Masalah umum dengan *use case* adalah dengan memfokuskan pada interaksi diantara dua user dan sistem. Sering kali perubahan pada proses bisnis adalah jalan terbaik untuk menyelesaikan masalah. Oleh karena itu

use case perlu dibedakan menjadi sistem *use case* dan *business use case*. Sistem *use case* adalah sebut interaksi dengan *software*, sedangkan *business use case* lebih menekankan kepada bagaimana sebuah bisnis merespon ke pelanggan atau kejadian/*event*. Berikut contoh pada gambar 2.10.



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 15 Use Case Extension

5. Skenario *Use case* / Dokumentasi *Use Case*

Setiap *use case* harus dideskripsikan dalam dokumen yang disebut dengan dokumen *flow of event*. Dokumen/skenario ini mendefinisikan apa yang harus dilakukan oleh sistem ketika aktor mengaktifkan *use case*. Struktur dokumen/skenario *use scase* ini bisa bermacam-macam, tetapi umumnya deksripsi ini paling tidak harus mengandung[20].

- 1) *Brief Description* yaitu deskripsi singkat.
- 2) *Actor* yang terlibat.
- 3) *Precondition* yang penting bagi use case untuk memulai.
- 4) *Main flow* dari kejadian yang bisa dirinci lagi menjadi *sub flow* yang lebih kecil agar dokumen lebih mudah di baca dan dimengerti.
- 5) *Alternatif flow* untuk mendefinisikan situasi perkecualian
- 6) *Postcondition* yang menjelaskan *state* dari sistem setelah *use case* berakhir.

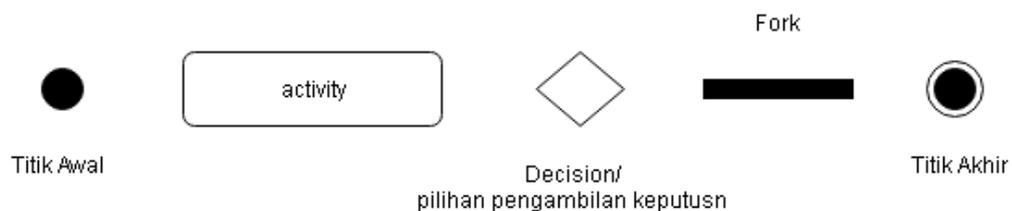
Tabel 2. 2 Contoh Dokumen/Skenario *use case*

Use Case	Nama use case
Brief Description	Deskripsi singkat
Actor	Aktor yang terlibat
Precondition	Yang penting bagi use case untuk memulai
Main flow	Kejadian yang bisa di rinci
Alternatif flow	Mendefinisikan situasi alternatifnya/pegecualian
Postcondition	Menjelaskan hasil dari <i>state</i> yang sudah dilakukan

2.14 Activity Diagram

Diagram aktivitas atau *activity* diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan workflow (aliran kerja) dalam banyak kasus. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa [20].

Dibawah ini adalah contoh beberapa simbol yang umum dipakai dan tentu digunakan dalam penelitian penulis, sebagai berikut [20].



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 16 Contoh simbol umum *activity* diagram

2.15 Sequence Diagram

Diagram sekuens digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini didalam *use case*.

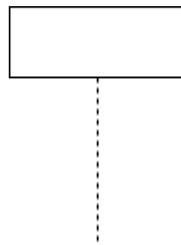
Komponen utama *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical* [20].

1) Obyek/*Participant*

Obyek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Pengertian

obyek hanya ada di UML 1, sedangkan di UML 2 istilah obyek diganti dengan *participant* [20].

Setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut aktivasi. Aktivasi mewakili sebuah eksekusi operasi dari obyek. Panjang kotak ini berbanding lurus dengan durasi aktivasi.



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 17 Obyek diagram *sequence*

2) *Message*

Sebuah *message* bisa jadi *simple*, *synchronous* atau *asynchronous*. *Message* yang simpel adalah sebuah perpindahan (transfer) kontrol dari satu *participant* ke *participant* yang lainnya. Jika sebuah atas *message* tersebut akan ditunggu sebelum diproses dengan urusannya. Namun jika *message asynchronous* yang dikirimkan, maka jawaban atas pesan pada diagram sekuens bisa dilihat sebagai berikut [20].

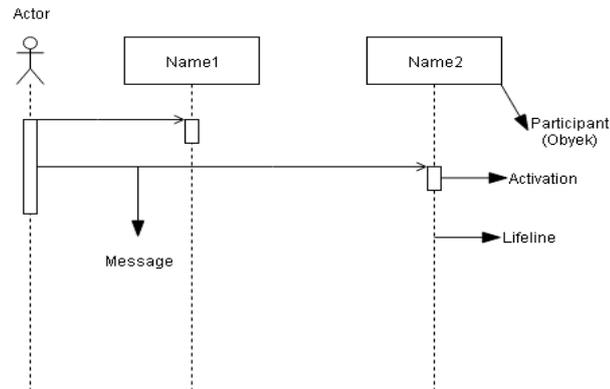


(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 18 Simbol-simbol *message*

3) *Time*

Time adalah diagram yang mewakili waktu pada arah *vertical*. Waktu dimulai dari atas kebawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat kebawah pengguna [20].



(Sumber : Pemodelan Visual dengan UML [20])

Gambar 2. 19 Simbol-simbol yang ada pada diagram *sequence*

Berikut Tabel 2.3 Operator umum yang sering digunakan pada *interaction frame*:

Tabel 2. 3 Operator-operator umum pada *interaction frame*

Operator	Keterangan
Alt	Alternatif fragment. Hanya kondisi True yang akan dijalankan
Opt	Optional fragment. Jika kondisi mendukungnya True
Par	Paralel. Setiap fragment dijalankan secara paralel
Loop	Looping. Dijalankan berulang kali, guard menunjukkan basis iterasi
Region	Fragment hanya punya satu thread untuk menjalankannya
Neg	Negative, fragment menunjukkan interaction yang salah
Ref	References, menunjukkan interaksi pada diagram class lain
Sd	Sequence diagram

