BAB 2
LANDASAN TEORI

2.1 Sejarah Prabu Siliwangi dan Harimau Prabu Siliwangi

Di Jawa Barat pada zaman dahulu kala ada sebuah Kerajaan Hindu yang besar dan cukup kuat, yaitu berpusat di kota Bogor. Kerajaan itu adalah Kerajaan "Pajajaran", pada saat itu raja yang memerintah yaitu Sri Baduga Maharaja atau yang terkenal disebut Prabu Siliwangi. Waktu mudanya Sri Baduga terkenal sebagai ksatria pemberani dan tangkas, bahkan satu-satunya yang pernah mengalahkan Ratu Japura (Amuk Murugul) waktu bersaing memperbutkan Subanglarang (istri kedua Prabu Siliwangi yang beragama Islam). Dalam berbagai hal, orang sezamannya teringat kepada kebesaran mendiang buyutnya (Prabu Maharaja Lingga Buana) yang gugur di Bubat yang digelari Prabu Wangi.

Dalam cerita "Moksanya Prabu Siliwangi" yang menyebutkan bahwa Prabu Siliwangi berubah menjadi seekor harimau.Hal ini terjadi karena Prabu Siliwangi menghindar dan tidak ingin memeluk agama Islam atas ajakan putranya,sehingga beliau lari ke hutan Sancang dan berubah menjadi Harimau [9]. Berikut ini penjelasan tentang Prabu Siliwangi, harimau, dan musuh-musuhnya.

2.1.1 Prabu Siliwangi

Prabu Siliwangi adalah seorang raja dari Kerajaan Padjadjaran. Prabu Siliwangi memiliki kekuatan yang hebat untuk melindungi kerajaan dari musuh-musuh jahat yang ingin menguasai Kerajaan Padjadjaran. Nama Siliwangi sudah tercatat dalam *Kropak 630* sebagai lakon pantun. Naskah itu ditulis tahun 1518 ketika Sri Baduga masih hidup. Lakon Prabu Siliwangi dalam berbagai versinya berintikan kisah tokoh ini menjadi raja di Pakuan atau yang lebih dikenal dengan kerajaan Padjadjaran. Peristiwa itu dari segi sejarah berarti saat Sri Baduga mempunyai kekuasaan yang sama besarnya dengan Wastu Kancana (kakeknya) alias Prabu Wangi (menurut

pandangan para pujangga Sunda). Menurut tradisi lama, orang segan atau tidak boleh menyebut gelar raja yang sesungguhnya, maka juru pantun memopulerkan sebutan Siliwangi. Dengan nama itulah ia dikenal dalam literatur Sunda [10]. Prabu Siliwangi mempunyai dua orang putra, dan satu orang putri yaitu Raden Walang Sungsang, atau dikenal dengan Pangeran Cakra Buana, lalu Nyi Mas Lara Santang dan yang ketiga bernama Raden Kian Santang [9]. Disamping itu, Prabu Siliwangi terkenal akan harimaunya yang sudah menjadi polemik di tanah sunda. Banyak sekali cerita tentang harimau Prabu Siliwangi di tanah sunda.

2.1.2 Harimau Prabu Siliwangi

Menurut cerita harimau Prabu Siliwangi, harimau Prabu Siliwangi merupakan jelmaan dari Prabu Siliwangi yang menghilang. Oleh karena itu, harimau selalu dikait-kaitkan dengan nama siliwangi. Cerita *"Moksanya Prabu Siliwangi"* ini adalah penjelmaan Prabu Siliwangi menjadi seekor harimau putih. Realitasnya, hingga kini masih banyak masyarakat Sunda (bahkan juga yang non-Sunda) meyakini metamorfosa Prabu Siliwangi menjadi harimau [9]. Selain itu, wangsit tersebut juga menjadi pedoman hidup bagi sebagian orang Sunda yang menganggap sifat-sifat *maung* seperti pemberani dan tegas, namun sangat menyayangi keluarga sebagai lelaku yang harus dijalani dalam kehidupan nyata. Sejatinya,sosok harimau sebagai jelmaan dari Prabu Siliwangi [10].

2.1.3 Jurig Ciwis

Jurig Ciwis merupakan suami Nyai Genit dan ayah Raden Adi Sijunjung [12]. Jurig Ciwis adalah tokoh jahat, yang juga merupakan musuh Prabu Silwangi. Jurig Ciwis prajurit atau anak buah dari Nini Durga, sama halnya dengan Patih Arga. Jurig Ciwis ingin menghalangi langkah Prabu Siliwangi untuk mengambil kerisnya yang telah dicuri oleh Patih Arga. Jurig Ciwis ingin membunuh dan mengambil kesaktian Maung Bodas, namun dicegah oleh Santang. Arga melepas kesaktian Jurig Ciwis. Maung

Bodas dituduh membunuh Jurig Ciwis oleh Arga. Arga menyarankan kepada Amuk Marugul untuk menculik Putri Selayati. Nyai Genit cs menyerang Rara Santang dan gurunya. Santang cs menyiapkan jebakan untuk pasukan Kuningan. Jebakan Santang terbaca oleh Nyai Genit dan prajuritnya. Nini Durga terluka terkena anak panah anak Syekh Nurdjati. Jurig Ciwis mengaku kepada Maung Bodas bahwa yang menyuruhnya adalah Amuk Marugul.

2.1.4 Patih Arga

Patih Arga merupakan mantan Senopati Pajajaran. Patih Arga memiliki kekasih yang bernama Embun Sari. Patih Arga terkenal dengan kelicikannya dan berupaya menghancurkan kerajaan dari politik adu domba. Ia dipengaruhi Nini Durga untuk melenyapkan kerajaan padjadjaran. Patih Arga diutus oleh Prabu Siliwangi untuk mencari sang putri. Ia tidak diperkenankan pulang jika tidak berhasil menemukan Rarasantang. Namun, usaha Patih Arga sia-sia belaka karenanya ia tidak berani pulang. Akhirnya, ia mengambil keputusan mengabdi di negeri Tajimalela[12].

2.1.5 Nini Durga

Musuh besar Pajajaran. Ia memiliki dendam pada Siliwangi karena anak buahnya dilenyapkan. Selain itu Nini Durga ingin menguasai kerajaan padjadjaran [12]. Nini Durga, tokoh aliran hitam. Tokoh ini sangat sakti, bisa menjelma jadi apa saja. Dia juga punya banyak pengikut yang sangat setia, rela melakukan apa saja yang diperintahkan Nini Durga. Nini Durga memiliki prajurit-prajtuirt yang jahat yang ingin menghancurkan Prabu Siliwangi dan kerajaan Padjadjaran. Banyak kejahatan dan kelicikan yang dilakukan Nini Durga terhadap Prabu Siliwangi maupun prajurit Padjadjaran agar Nini Durga bisa mengalahkan Prabu Siliwangi dan menguasai kerajaan Padjadjaran.

2.2 Multimedia

Arti multimedia berbeda dari sudut pandang orang yang berbeda. Secara umum, multimedia berhubungan dengan penggunaan lebih dari satu macam cara untuk menyajikan informasi.

Dalam dunia komputer, multimedia berhubugan dengan perangkat lunak yang digunakan dalam pengembangan dengan lebih dari satu cara untuk menyampaikan informasi kepada pengguna, seperti teks dan suara. Secara teknis, multimedia dalam komputer bukan merupakan hal baru, karena integrasi antara teks dan gambar merupakan komponen utama dalam banyak perangkat lunak. Pada saat ini, multimedia mempunyai arti tidak hanya integrasi antara teks dan grafik sederhana saja, tetapi dilengkapi dengan suara dan animasi [9].

Multimedia terdiri dari beberapa komponen, yaitu [9]:

1. *Teks* : Teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia.
2. Grafik : Secara umum grafik berarti bahwa gambar garis. Manusia sangat berorientasi pada visual, dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi.
3. *Images*

   Secara umum images berarti gambar raster (*halftone drawing)*, seperti gambar.
4. Animasi

   Animasi menggambarkan objek yang bergerak agar kelihatan hidup.
5. Suara

   Suara dapat lebih memperjelas pengertian yang ditampilkan dengan cara lain.
6. *Interactive link*

   Sebagian dari multimedia adalah interaktif, dimana pengguna dapat menggunakan *keyboard* untuk melakukan perintah.

2.3 Game Fighting

Game fighting dikelompokan kedalam bagian aksi. *Game fighting* merupakan *game* yang intinya harus menjatuhkan lawan, baik dengan pukulan, tendangan, jurus, maupun combo. Jenis *genre game* ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, pengenalan karakter dan timing sangatlah penting untuk memainkannya. Inti dari *game fighting* adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya). Contoh *game fighting* diantaranya *Bloody Roar, Street Fighter, Tekken, Mortal Kombat, Soul Calibur* dan *King of Fighter*.

2.4 Game

Game merupakan kata dalam bahasa inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*. Suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi strategi yang rasional. Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi :

"Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi." (Theory of Games and Economic Behavior (3d ed. 1953)) [7].

Game bukanlah hal yang tabu di mata masyarakat. Hampir semua orang dari berbagai kalangan pernah memainkan game. Mendefinisikan apakah yang dimaksud dengan game, tidak cukup dengan hanya melihat kamus bahasa. Makna sekilas dari game memberikan pengertian bahwa

game merupakan suatu aktifitas yang tidak dilakukan dengan sungguh-sungguh. Untuk mengetahui apa yang sesungguhnya disebut dengan game, maka paling tidak kita dapat memahaminya dari adanya sejumlah pengertian game yang biasa kita alami dalam kehidupan.

Terdapat 5 kategori istilah *game*, yaitu :

1. *Board Games* (Permainan Papan) Game pada katagori ini membutuhkan suatu papan yang terbagi dalam sektor-sektor tertentu (dengan garis-garis) dan didalamnya terdapat sejumlah alat main yang dapat digerakkan. Termasuk dalam katagori ini adalah catur. Dua buah pemain akan berhadapan dan saling mengadu strategi sesuai dengan aturan untuk mencapai daerah lawan atau mempertahankan daerahnya sendiri, mengalahkan bidak musuh, mengumpulkan sesuatu. Pemain pada board games ini akan berusaha menganalisis hubungan-hubungan geometri yang ada pada papan dan bidak.
2. *Card Games* (Permainan Kartu) Games ini akan memanfaatkan simbol dari 52 kartu yang terbagi dalam dua faktor : suit (4 nilai) dan rank (13 nilai). Permainan akan dilakukan sekitar bagaimana membuat kombinasi dari 52 kartu tersebut. Sejumlah ketentuan dibuat untuk mengatur bagaimana cara-cara untuk membuat kombinasi tersebut. Permainan kartu bridge/truf termasuk kelompok games ini.
3. *Athletic Games* (Permainan Atletik) Permainan games jenis ini lebih cenderung pada penggunaan fisik daripada mental. Aturan game dibuat dengan keharusan pemain untuk melakukan sejumlah aksi tertentu. Hal yang terkait dengan kekuatan badan, kecepatan, ketepatan dan kerjasama menjadi bagian utama dari game atletik. Dalam hal ini harus dibedakan antara game dengan kompetisi. Kompetisi tidak mengharuskan adanya kerjasama dan pemain bekerja secara individu. Dua orang yang beradu lari bukanlah termasuk game tetapi kompetisi. Perbedaan utama antara game dan kompetisi adalah dalam hal interaksi diantara peserta. Pada kompetisi tidak terjadi interaksi diantara pemain, kompetisi yang membolehkan adanya interaksi diantara pemainnya termasuk game.

4. *Children Games* (Permainan Anak) Aktifitas seperti berlari, sembunyi, melempar dan menangkap adalah menjadi ciri utama game anak-anak. Umumnya game ini menekankan pada aktifitas kelompok sebagai latihan untuk berkehidupan sosial. Walaupun dalam game ini terdapat juga upaya untuk saling mengalahkan secara mental atau fisik namun tujuan utamanya bukanlah untuk meraih kemenangan satu diatas yang lain tetapi sebagai ilustrasi kerjasama dalam kehidupan manusia. Penggunaan sejumlah alat untuk membantu ilustrasi dapat meningkatkan improvisasi game dan meningkatkan keterlibatan yang lebih baik dari pemain.
5. *Computer Games* (Permainan Komputer) Game ini dimainkan lewat bantuan alat komputer. Terdapat 5 alat yang dapat dikategorikan sebagai komputer, yaitu :
   a. *Expensive dedicated machine,* mesin yang dioperasikan dengan koin untuk memainkankannya.
   b. *Inexpensive dedicated machine,* disebut juga dengan *hand held machine*. Alat game watch termasuk dalam katagori ini.
   c. *Multiprogram home,* mesin seperti Atari, Nintendo termasuk dalam kelompok komputer ini
   d. Personal computer
   e. Mainframe computer

Komputer game berbeda dengan jenis game yang lain karena tidak ada pergerakan secara fisik atau interaksi langsung dengan object kecuali lewat perantaraan komputer. *Software* yang dibuat harus dapat menangkap reaksi yang cepat dari interaksi yang dihasilkan dengan pemain. Karena itu *software* untuk computer games harus bersifat real time. Kompleksitas game adalah bergantung dari kemampuan merepresentasikan aturan dan lingkungan game dalam program yang dibuat.

Diantara sekian banyak definisi game, maka definisi yang umum untuk computer game adalah game Komputer adalah sebuah program software dimana satu atau lebih pemain berusaha untuk membuat keputusan

lewat kontrol terhadap object dan resource guna memenuhi satu tujuan tertentu.

2.4.1 Sejarah Game Pertama di Indonesia

Divinekids adalah *game* yang diakui dan mendapat award resmi dari MURI (Museum Rekor Indonesia) sebagai Game Pertama Indonesia pada tahun 2004 dan banyak majalah-majalah Game dan Komputer (Gamestation, PCGAME, Bobo, INDOPOS, Game21 dan lain sebagainya). Sebenarnya sudah banyak game-game buatan orang-orang indonesia yang lain tetapi karena kekurangan syarat-syarat minimal tertentu seperti : Karakter menggunakan karakter asing, penyebaran yang kurang luas, bahasa yang digunakan bukan bahasa Indonesia, tidak terdaftar resmi Copyright ataupun Trademarknya dan lain sebagainya, karena itu tidak dipilih sebagai Game Pertama Indonesia Resmi.

Menurut sebagian pengamat game ini adalah kurangnya investigasi MURI dalam hal melihat adanya game-game tersebut sebelum memberikan gelar game pertama ke David Setiabudi. Terlebih mereka tidak meminta pendapat dari pihak yang memahami dunia game Indonesia itu sendiri.

Pengamat-pengamat yang berpendapat berbeda dengan MURI dan majalah-majalah PC dan Game, memposisikan sejarah game Indonesia sebagai berikut: Sejarah Game Indonesia menurut Gamedevid. Selain itu juga masih ada banyak game-game lain yang belum terdokumentasi dengan baik. Beberapa game buatan Elex Media Komputindo dengan copyright resmi dan penyebaran yang luas ada juga yang dibuat sebelum sebelum dirilisnya Divine Kids. Game ini dibuat dan dibagikan secara gratis melalui CD majalah2 game/ komputer dan melalui Internet.

Game-game Divinekids berjumlah 38 lebih dan terus bertambah (2006). Pembuatnya adalah David Setiabudi dan mahasiswa-mahasiswi bimbingannya. Pembuatan musik gamenya dibuat oleh Rama Aurora, tunanetra berbakat membuat musik digital.

David Setiabudi membuat game juga tahun 1992 tetapi tidak diakui sebagai game pertama Indonesia karena kurang memenuhi syarat. Ada juga game-game buatan Korea yang terkenal diterjemahkan ke bahasa Indonesia & game-game yang penyebaran kurang luas & game yang berkaraker asing atau berbahasa asing & tidak terdaftar Copyright secara resmi tidak terpilih menjadi Game Pertama Indonesia. Ada yang berpendapat asal tahun tertulis paling tua itu patut disebut Game Pertama Indonesia tidak perduli dengan buatan negara mana, atau tersebar di lokal saja. Sampai sekarang masih diperdebatkan kelompok tertentu [5].

2.4.2 Pengertian Game

Pengertian game menurut beberapa ahli [5]:

1. Menurut Agustinus Nilwan dalam bukunya Pemrograman Animasi dan Game Profesional terbitan Elex Media Komputindo, *game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami pengunaan animasi haruslah memahami pembuatan game. Atau jika ingin membuat *game*, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.
2. Menurut Clark C. Abt, Game adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi oleh konteks tertentu (misalnya, dibatasi oleh peraturan).
3. Menurut Chris Crawford, seorang *computer game designer* mengemukakan bahwa *game*, pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada pelaku aktif *(player),* ada pelaku pasif *(NPC*).
4. Menurut David Parlett, Game adalah sesuatu yang memiliki "akhir dan cara mencapainya" : artinya ada tujuan, hasil dan serangkaian peraturan untuk mencapai keduanya.
5. Menurut Roger Caillois, seorang sosiolog Perancis, dalam bukunya yang berjudul Les jeux et les hommes menyatakan game adalah aktivitas yang mencakup karakteristik berikut: *fun* (bebas bermain adalah pilihan bukan

kewajiban), *separate* (terpisah), *uncertain, non-productive, governed by rules* (ada aturan), *fictitious* (pura-pura).

6. Menurut Greg Costikyan, Game adalah “sebentuk karya seni di mana peserta, yang disebut Pemain, membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan”.

2.4.3 Klasifikasi *Game*

2.4.3.1 Berdasarkan Jenis *Platform* yang digunakan

Berikut klasifikasi *game* berdasarkan jenis platform yang digunakan :

1. *Arcade games,* yaitu yang sering disebut *ding-dong* di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis video games tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya)
2. *PC Games*, yaitu video game yang dimainkan menggunakan Personal Computers.
3. *Console games*, yaitu video games yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. *Handheld games,* yaitu yang dimainkan di console khusus video game yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games,* yaitu yang dapat dimainkan atau khusus untuk mobilephone atau PDA.

2.4.3.2 Berdasarkan Genre permainannya

Berikut klasifikasi *game* berdasarkan genre permainannya :[5]

1. Aksi *Shooting*, video game jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari game jenis ini adalah tembak, tembak dan tembak. Termasuk didalam-nya :

a. *First person shooting (FPS)* seperti *Counter Strike* dan *Call of Duty* Drive n shoot, menggunakan unsur simulasi kendaraan tetapi tetap dengan tujuan utama menembak dan menghancurkan lawan, contoh : *Spy Hunter, Rock and Roll Racing, Road Rash.*
b. *Shoot em up,* seperti Raiden, 1942, dan gradius.
c. Beat em up (tonjok hajar) seperti *Double Dragon* dan *Final Fight*, lalu *hack and slash* (tusuk tebas) seperti *Shinobi* dan *Legend of Kage.*
d. *Light gun shooting,* yang menggunakan alat yang umumnya berbentuk seperti senjata, seperti *Virtua Cop* dan *Time Crisis.*
e. *Fighting* (pertarungan) *Game fighting* dikelompokan kedalam bagian aksi. *Game fighting* merupakan *game* yang intinya harus menjatuhkan lawan, baik dengan pukulan, tendangan, jurus, maupun combo. Jenis *genre game* ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, pengenalan karakter dan timing sangatlah penting untuk memainkannya. Inti dari *game fighting* adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya). Contoh *game fighting* diantaranya *Bloody Roar, Street Fighter, Tekken, Mortal Kombat, Soul Calibur* dan *King of Fighter.*

2. Petualangan. Bedanya dengan jenis video game aksi-petualangan, *refleks* dan kelihaian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini. *Video Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara *visual*, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat.
3. Simulasi, Konstruksi dan manajemen. *Video Game* jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: *Sim City, The Sims, Tamagotchi.*

4. *Role Playing. Video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana siring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter. seperti *Final Fantasy, Dragon Quest* dan *Xenogears. Never Winter Nights, baldurs gate, Elder Scroll,* dan *Fallout.*
5. *Puzzle. Video game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam video game petualangan maupun *game* edukasi. *Tetris, Minesweeper, Bejeweled, Sokoban* dan *Bomberman.*
6. Simulasi kendaraan. Video Game jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, muskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.
7. Olahraga. Singkat padat jelas, bermain sport di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada yang menambah unsur fiksi seperti *NBA JAM*. Contohnya pun jelas, Seri *Winning Eleven*, seri *NBA,* seri *FIFA, John Madden NFL, Lakers vs Celtics, Tony hawk pro skater,* dll [5].

2.5 OOP (Object Oriented Programming)

OOP (*Object Oriented Programming*) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang

berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam *kelas-kelas* atau *objek-objek.*

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep berikut:

Kelas (*Class*) — kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*class* of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah *class* sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak

menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

Objek (*Object*) - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

Abstraksi (*Abstract*) - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

Enkapsulasi (*Encapsulation*) - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

Polimorfisme (*Polimorfism*) melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan

hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai *polimorfisme* melalui penggunaan fungsi kelas-pertama.

Inheritas (*Inheritance*). Mengatur *polimorfisme* dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas) [4].

2.6 Control Class

Kelas-kelas kendali (*control class*) merepresentasikan koordinasi, urutan, dan kendali objek lain dan digunakan untuk mengendalikan hubungan dengan suatu *use case* tertentu[6].

2.7 Boundary Class

Kelas pembatas (*boundary class*) digunakan untuk memodelkan interaksi antara system/perangkat lunak dengan actornya (misalnya interaksi antara system/perangkat lunak dengan pengguna system/perangkat lunak dan dengan system eksternal)[6].

2.9 Entity Class

Kelas Entitas (*entity class*) digunakan untuk memodelkan informasi yang berumur realtif pangjang dalam system (*persistent class*). Kelas-kelas entitas memodelkan informasi dan perilaku terkait dari sejumlah konsep penting seperti individu, objek dunia nyata, atau event yang terjadi di dunia nyata[6].

2.10 *UML* (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem (Flowler, 2006). *Unified Modeling Language* (*UML*) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. *UML* adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. *UML* mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang *UML* sudah mulai banyak digunakan oleh para praktisi OOP.

UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM [10].



Gambar 2.2 Logo UML

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. *UML* dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian *UML* dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan *UML* dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri

peranti lunak dan pengembangan sistem.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. ***Use Case Diagram*** untuk memodelkan proses bisnis.
2. ***Conceptual Diagram*** untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. ***Sequence Diagram*** untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. ***Collaboration Diagram*** untuk memodelkan interaksi antar *objects*.
5. ***State Diagram*** untuk memodelkan perilaku *objects* di dalam sistem.
6. ***Activity Diagram*** untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
7. ***Class Diagram*** untuk memodelkan struktur kelas.
8. ***Object Diagram*** untuk memodelkan struktur *object*.
9. ***Component Diagram*** untuk memodelkan komponen *object*.
10. ***Deployment Diagram*** untuk memodelkan distribusi aplikasi.

2.11 Java

Java adalah bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya seperti Patrick Naugton, Chris Warth, Ed Frank, dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama Susn Microsystems, pada tahun 1991 (Raharjo.B, Heryanto. I, Arif Haryano, 2009, hlm. 1-2). Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun oada tahun 1995 diganti namanya menjadi “Java”.

Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan diberbagai macam perangkat elektronik, seperti *microwave oven* dan *remote control*, sehingga Java harus bersifat *porTabel* atau yang sering disebut dengan *platform independent* (tidak bergantung pada *platform*). Itula yang menyebabkan dalam dunia pemrograman Java, dikenal adanya istilah ‘*write once, run everywhere*’,

yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah *platform* manapun, tanpa harus melakukan perubahan kode program.

2.11.1 Java Versi 1

Pada awalnya perilisannya, versi Java masih disebut dengan JDK (*Java Development Kit*). Dalam JDK, semua kebutuhan untuk pengembangan program dan eksekusi program masih tergabung jadi satu. Penamaan ini berlaku sampai dengan Java 1.1. Namun sekarang, setelah Java 1.2, Sun Mycrosystems menamainya dengan JSDK (*Java Software Development Kit*) dalam hal ini kebutuhan untuk pengembangan program dipisahkan dengan kebutuhan eksekusi. Bagian *software* yang digunakan untuk kebutuhan eksekusi program disebut dengan JRE (*Java-Runtime Environment*). Selanjutnya, Java 1.2 disederhanakan penamaannya menjadi "Java 2" [4].

2.11.2 Java Versi 2

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2 yaitu sebagai berikut : [4]

1. **Java 2 Standard Edition (J2SE),** yang digunakan untuk mengembangkan aplikasi-aplikasi *desktop* dan *applet* (aplikasi untuk Java yang dapal dijalankan di dalam *browser web*).
2. **Java 2 Enterprise Edition (J2EE),** merupakan *superset* dari J2SE yang memperbolehkan kita untuk mengembangkan aplikasi-aplikasi berskala besar(*enterprise*), yaitu dengan melakukan pembuatan aplikasi-aplikasi di sisi *server* dengan menggunakan EJBs (*Enterprise JavaBeans*), aplikasi *web* dengan menggunakan *Servlet* dan JSP (*JavaServer Pages*) dan teknologi lainnya seperti CORBA (*Common Object Request Broker Architecture*) dan XML (*Extensible Markup Language*).
3. **Java 2 Micro Edition (J2ME),** merupakan subset dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE

secara penuh seperti pada teknologi *mobile.*

2.12 Greenfoot

Greenfoot merupakan sebuah IDE berbasis java yang dikhususkan untuk pembuatan game sederhana. Greenfoot diprakarsai oleh Michael Kolliung pada tahun 2003. Prototype yang pertama dibangun oleh Poul Henriksson dan Michael Kolling pada tahun 2003/2004. semenjak tahun 2005 pembuatan greenfoot ini dilanjutkan dengan melibatkan anggota kelompok dari University of Kent dan Deakin University.

Pada awalnya, greenfoot bertujuan untuk menarik minat anak-anak untuk belajar pemrograman, greenfoot ini cocok untuk anak usia 13 tahun ke atas. Cara penggunaannya pun cukup mudah. Selain berbentuk teks seperti editor lainnya, pada greenfoot ini terdapat perangkat alat bantu seperti *class* browser, editor dan compiler. Greenfoot juga mendukung bahasa Java secara utuh. Dengan adanya bantuan perangkat-perangkat tadi, konsep OOP pada greenfoot lebih mudah dimengerti.

Didalam greenfoot terdapat project yang dinamakan scenario. Di dalam project tersebut, kita bisa membuat dua macam tipe *class* dari *library* greenfoot. Yaitu *class world* dan *actor*. *World* merupakan latar dari game yang akan dibuat. Untuk membuat *class world,* caranya adalah dengan meng-klik *new sub class* di bagian *world*. Maka akan mucul sebuah kotak dialog. Kemudian kita diminta untuk memberi nama untuk kelas *world* tersebut. Setelah itu kita dapat memilih gambar sebagai latar belakang *class* world yang akan kita buat. Begitu pula pada *Class actor,* membuat actor caranya adalah dengan meng-klik icon aktor, kemudian pilih *new subclass.*

Pada dasarnya, Greenfoot dapat dipandang sebagai kombinasi dari :

1. Sebuah *framework* untuk membuat skema program Java yang dapat divisualisasikan dalam bentuk dua dimensi.
2. Sebuah IDE (*Integrated Development Environment*), termasuk *browser* kelas, *editor* program, *compiler,* antarmuka untuk eksekusi, dan lain-lain.

Seperti yang telah diutarakan, framework Greeenfot dapat digunakan untuk program-program yang dapat divisualisasikan dalam grid dua dimensi.

Framework ini mempunyai dua fungsi utama :

1. Mempermudah representasi objek secara grafis.
2. Menyediakan fasilitas dan antarmuka untuk mengkontrol eksekusi program (*start, stop, step*).

Framework Greenfoot mempunyai sebuah kelas basis bernama GreenfootObject yang harus menjadi induk dari semua kelas yang ingin "menampakkan dirinya" secara visual. Kelas ini mempunyai dua metode penting yang harus diimplementasikan :

1. Metode getImage() yang harus memberikan nilai balik gambar/*image* dari sebuah objek; dan
2. Metode act() yang akan dipanggil dalam setiap langkah (*step*) dari proses iterasi (*simulation loop*).

Dalam Greenfoot terdapat beberapa istilah yang dipergunakan, diantaranya skenario atau projek. Skenario merupakan sekumpulan kelas mengimplementasikan kelas-kelas basis untuk aplikasi tertentu. Idenya adalah, supaya pengguna Greenfoot mempunyai sesuatu yang dapat digunakan sebagai pijakan awal untuk bereksperimen, untuk melakukan modifikasi atau untuk melakukan penambahan fungsionalitas, seiring dengan kemajuan pemakai dalam memahami bahasa Java.

Adapula IDE Greenfoot yang terdiri atas beberapa elemen yang biasa ditemui dalam sebuah IDE pada umumnya, misalnya :

1. Editor kode sumber
2. Browser kelas
3. Fasilitas kompilasi
4. Kontrol eksekusi program
5. Debugger

Greenfoot dapat memvisualisasikan objek-objek dari sebuah scenario dan menyediakan sebuah bentuk antarmuka bagi pengguna untuk berinteraksi dengan objek-objek tersebut. Selain itu, juga tersedia fasilitas untuk mengontrol eksekusi program dari scenario yang sedang berjalan.

Fitur penting dari Greenfoot yanag membedakan dengan IDE yang lain adalah interaksi langsung. Interaksi dapat dilakukan baik untuk kelas maupun untuk objek. Misalnya :

1. Untuk menciptakan sebuah objek, kita bisa memilih secara langsung item menu konteks (*context-menu / pop-up menu*) pada sebuah kelas untuk membentuk objek baru.
2. Setelah objek diinisiasi, ia dapat ditempatkan pada lingkungan dengan mengklik pada posisi yang diinginkan.

   Interaksi objek juga dapat dilakukan dengan cara yang hamper serupa :

1. Objek yang telah eksis bisa dipindahkan setiap saat dengan melakukan *drag* (menekan tombol kiri *mouse*, menggeser dan melepaskan tombol setelah sampai pada posisi yang diinginkan) pada objek tersebut.
2. Pilih/panggil metode yang diinginkan dari menu konteks, dengan mengklik kanan objek tersebut. Jika metode yang dipanggil mengubah keadaan (*state*) objek yang divisualisasikan, maka perubahan akan segera ditampilkan [3].

2.13 Pengujian Perangkat Lunak

2.13.1 Pengujian Alpha

2.13.1.1 Pengertian Pengujian

Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan design kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak adalah sebagai berikut[11] :

1. Pengujian dimulai dari level komponen hingga integrasi antarkomponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*independent Test Group* (ITG)).
4. Pengujian dan penirkutuan (*debugging*) merupakan aktifitas yang berbeda, tapi penirkutuan (*debugging*) harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih fokus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengembang tanpa harus menemukan lokasi kesalahan pada kode program. Penirkutuan (*debugging*) adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga dapat segera diperbaiki oleh pembuat program (*programmer*).

Pengujian untuk validasi memiliki pendekatan sebagai berikut[1] :

1. *Black-Box Testing* (Pengujian kotak hitam)

Yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dank ode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

a. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.

b. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.13.2 Pengujian Beta

2.13.3 Pengertian kuisioner

Menurut Sugiyono (2006, p.135), kuesioner adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya.

Menurut Sekaran (2006, p.82), kuesioner adalah daftar pertanyaan tertulis yang telah dirumuskan sebelumnya yang akan responden jawab, biasanya dalam alternative yang didefinisikan dengan jelas. Kuesioner merupakan suatu mekanisme pengumpulan data yang efisien jika peneliti mengetahui dengan tepat apa yang diperlukan dan bagaimana mengukur variabel penelitian. Kuesioner dapat diberikan secara pribadi, disuratkan kepada responden, atau disebarkan secara elektronik.

Menurut Hasan (2002, p.83), kuesioner (angket) adalah teknik pengumpulan data dengan menyerahkan atau mengirimkan daftar pertanyaan untuk diisi oleh responden. Dari definisi-definisi diatas dapat disimpulkan bahwa kuesioner adalah suatu teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis yang sudah dirumuskan sebelumnya kepada responden untuk dijawab.

2.13.3.1 Isi Dari Kuesioner

Menurut Nazir (2003, p.203-204), isi dari kuesioner secara umum dapat berupa :

1. Pertanyaan tentang fakta

Pertanyaan tentang fakta-fakta yang dianggap dikuasai oleh responden. Fakta-fakta tersebut bisa saja berhubungan dengan responden, dengan suatu keadaan ataupun dengan orang-orang yang dikenal oleh responden sendiri.

2. Pertanyaan tentang pendapat (opinion)

Pertanyaan mengenai pendapat berupa suatu keadaan atau suatu situasi. Jawaban pertanyaan tentang pendapat pada umumnya bersifat laten dan baru muncul bila ditanyakan. Pertanyaan mengenai pendapat banyak sekali seginya, menyangkut masalah moral, kebudayaan, harga diri dan sebagainya.

3. Pertanyaan tentang persepsi diri

Pertanyaan yang meliputi cara responden menilai sesuatu tentang perilakunya sendiri dalam hubungannya dengan orang lain atau lingkungan.

2.13.3.2 Cara Mengungkapkan Pertanyaan

Menurut Nazir (2003, p.205-206), cara mengungkapkan pertanyaan adalah sebagai berikut :

1. Perkataan dan kalimat harus sederhana.

Gunakan kalimat dan kata-kata yang sederhana. Hindarkan pemilihan kata-kata sulit.

2. Pertanyaan sebaiknya khas.

Pertanyaan janganlah terlalu umum, tetapi sebaiknya spesifik dan khas.

3. Pertanyaan jangan berarti dua.

Jika pertanyaan dapat mendua artinya, maka tiap orang akan mengartikan pertanyaan tersebut dengan pengertian yang berbeda.

4. Jangan gunakan kata yang samar-samar artinya.

Hindarkan kata-kata yang artinya samar-samar. Kata-kata ini dapat menghasilkan jawaban samar-samar pula. Kata-kata : banyak, secara keseluruhan, jenis,biasa, agak, dan sebagainya merupakan kata-kata yang samar-samar.

5. Pertanyaan yang mengandung sugesti.

Pertanyaan, baik karena isi atau kata-kata yang digunakan, dapat menjuruskan responden kepada suatu jawaban tertentu. Pertanyaan seperti ini dinamakan pertanyaan yang memberikan sugesti.

6. Pertanyaan presumasi.

Pertanyaan presumasi adalah pertanyaan yang bersandar kepada anggapan bahwa responden termasuk dalam kategori yang mempunyai sifat ingin ditanyakan, ataupun responden mempunyai pengetahuan yang baik tentang kelompok yang ingin ditanyakan.

7. Pertanyaan yang membuat seseorang malu.

Hindarkan pertanyaan yang membuat malu atau terlalu pribadi bagi responden.

8. Pertanyaan yang mengundang ingatan kuat.

Hindarkan pertanyaan-pertanyaan yang memerlukan ingatan yang kuat dari responden.

2.13.3.3 Jenis Pertanyaan / Kuesioner

Menurut Hasan (2002, p.84-85), kuesioner (angket) dibedakan atas 3 golongan adalah sebagai berikut :

1. Angket terbuka (opened questionare)

Merupakan angket yang pertanyaan atau pernyataannya memberikan kebebasan kepada responden, untuk memberikan jawaban dan pendapatnya sesuai dengan keinginan mereka.

2. Angket tertutup (closed questionare)

Merupakan angket yang pertanyaan atau pernyataannya tidak memberikan kebebasan kepada responden, untuk memberikan jawaban dan pendapatnya sesuai dengan keinginan mereka.

3. Angket semi terbuka (semi opened questionare)

Merupakan angket yang pertanyaan atau pernyataannya memberikan kebebasan kepada responden untuk memberikan jawaban dan pendapat menurut pilihan-pilihan jawaban yang telah disediakan.

2.13.4 Skala Likert

Skala *Likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial, fenomena sosial

ini telah ditetapkan secara spesifik oleh peneliti dan selanjutnya disebut sebagai variable penelitian[16].

Dengan menggunakan skala *likert* variable yang akan diukur dijabarkan menjadi indikator variable. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan.

Jawaban setiap instrument yang menggunakan skala *likert* mempunyai gradasi dari yang sangat positif hingga yang sangat negatif, yang dapat berupa kata – kata antara lain :

a. Sangat Setuju	a. Selalu
b. Setuju	b. Sering
c. Ragu – ragu	c. Kadang - kadang
d. Tidak Setuju	d. Tidak Pernah
e. Sangat Tidak Setuju	

a. Sangat Positif	a. Sangat Baik
b. Positif	b. Baik
c. Negatif	c. Tidak Baik
d. Sangat Negatif	d. Sangat Tidak Baik

Untuk keperluan analisis kuantitatif, maka jawaban itu dapat diberi skor, misalkan:

1. Setuju/selalu/sangat positif diberi skor 5
2. Setuju/sering/positif diberi skor 4
3. Ragu-ragu/kadang-kadang/netral diberi skor 3
4. Tidak setuju/hapir tidak pernah/negative diberi skor 2
5. Sangat tidak setuju/tidak pernah 1

Instrument penelitian yang menggunakan skala likert dapat dibuat dalam bentuk ceklis atau pilihan ganda.

Contoh bentuk pilihan ganda :

Berilah salah satu jawaban terhadap pertanyaan berikut sesuai dengan pendapat anda, dengan cara memberi tanda lingkaran pada jawaban yang tersedia.

Prosedur kerja yang baru itu akan segera diterapkan dilembaga anda?

a. Sangat setuju
b. Setuju
c. Ragu-ragu/netral
d. Tidak setuju
e. Sangat tidak setuju

SS = Sangat setuju	diberi skor	5
ST = Setuju	diberi skor	4
RG = Ragu-ragu	diberi skor	3
TS = Tidak setuju	diberi skor	2
STS = Sangat tidak setuju	diberi skor	1

Kemudian dengan teknik pengumpulan data angket, maka instrument tersebut misalnya diberikan kepada 100 orang karyawan yang diambil secara random. Dari 100 orang pegawai setelah dilakukan analisis misalnya :

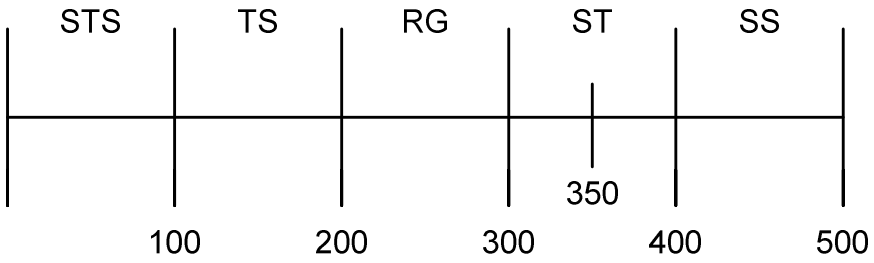
25	orang menjawab	SS
40	orang menjawab	ST
5	orang menjawab	RG
20	orang menjawab	TS
10	orang menjawab	STS

Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut :

Jumlah skor untuk 25 orang yang menjawab SS	=	25 x 5	=	125
Jumlah skor untuk 40 orang yang menjawab ST	=	40 x 4	=	160
Jumlah skor untuk 5 orang yang menjawab ST	=	5 x 3	=	15
Jumlah skor untuk 20 orang yang menjawab TS	=	20 x 2	=	40
Jumlah skor untuk 10 orang yang menjawab STS	=	10 x 1	=	10
Jumlah total			=	350

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian = 350. Jadi berdasarkan data itu maka tingkat persetujuan terhadap metode kerja baru itu = (350 : 500) x 100% = 70% dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Gambar 2. 1 Gambar kontinum dari contoh skala *likert*

Jadi berdasarkan data yang diperoleh dari 100 responden maka rata-rata 350 terletak pada daerah setuju.