BAB 2
LANDASAN TEORI

2.1 Pengertian Banjir

Banjir merupakan fenomena alam yang biasa terjadi di suatu kawasan yang banyak dialiri oleh aliran sungai. Secara sederhana banjir dapat didefinisikan sebagainya hadirnya air di suatu kawasan luas sehingga menutupi permukaan bumi kawasan tersebut.

Dalam cakupan pembicaraan yang luas, kita bisa melihat banjir sebagai suatu bagian dari siklus hidrologi, yaitu pada bagian air di permukaan Bumi yang bergerak ke laut. Dalam siklus hidrologi kita dapat melihat bahwa volume air yang mengalir di permukaan Bumi dominan ditentukan oleh tingkat curah hujan, dan tingkat peresapan air ke dalam tanah.

Aliran Permukaan = Curah Hujan – (Resapan ke dalam tanah + Penguapan ke udara)

Air hujan sampai di permukaan Bumi dan mengalir di permukaan Bumi, bergerak menuju ke laut dengan membentuk alur-alur sungai. Alur-alur sungai ini di mulai di daerah yang tertinggi di suatu kawasan, bisa daerah pegunungan, gunung atau perbukitan, dan berakhir di tepi pantai ketika aliran air masuk ke laut. Secara sederhana, segmen aliran sungai itu dapat kita bedakan menjadi daerah hulu, tengah dan hilir.

1. *Daerah hulu:* terdapat di daerah pegunungan, gunung atau perbukitan. Lembah sungai sempit dan potongan melintangnya berbentuk huruf “V”. Di dalam alur sungai banyak batu yang berukuran besar (bongkah) dari runtuhan tebing, dan aliran air sungai mengalir di sela-sela batu-batu tersebut. Air sungai relatif sedikit. Tebing sungai sangat tinggi. Terjadi erosi pada arah vertikal yang dominan oleh aliran air sungai.
2. *Daerah tengah:* umumnya merupakan daerah kaki pegunungan, kaki gunung atau kaki bukit. Alur sungai melebar dan potongan melintangnya berbentuk huruf “U”. Tebing sungai tinggi. Terjadi erosi pada arah horizontal, mengerosi batuan induk. Dasar alur sungai melebar, dan di

dasar alur sungai terdapat endapan sungai yang berukuran butir kasar. Bila debit air meningkat, aliran air dapat naik dan menutupi endapan sungai yang di dalam alur, tetapi air sungai tidak melewati tebing sungai dan keluar dari alur sungai.

3. *Daerah hilir:* umumnya merupakan daerah dataran. Alur sungai lebar dan bisa sangat lebar dengan tebing sungai yang relatif sangat rendah dibandingkan lebar alur. Alur sungai dapat berkelok-kelok seperti huruf "S" yang dikenal sebagai "meander". Di kiri dan kanan alur terdapat dataran yang secara teratur akan tergenang oleh air sungai yang meluap, sehingga dikenal sebagai "dataran banjir". Di segmen ini terjadi pengendapan di kiri dan kanan alur sungai pada saat banjir yang menghasilkan dataran banjir. Terjadi erosi horizontal yang mengerosi endapan sungai itu sendiri yang diendapkan sebelumnya.

Dari karakter segmen-segmen aliran sungai itu, maka dapat dikatakan bahwa :

1. Banjir merupakan bagian proses pembentukan daratan oleh aliran sungai. Dengan banjir, sedimen diendapkan di atas daratan. Bila muatan sedimen sangat banyak, maka pembentukan daratan juga terjadi di laut di depan muara sungai yang dikenal sebagai "delta sungai."
2. Banjir yang meluas hanya terjadi di daerah hilir dari suatu aliran dan melanda dataran di kiri dan kanan aliran sungai. Di daerah tengah, banjir hanya terjadi di dalam alur sungai.

Dari pengertian di atas dapat disimpulkan bahwa banjir adalah peristiwa yang terjadi ketika aliran air yang berlebihan merendam daratan. Banjir juga dapat terjadi di sungai, ketika alirannya melebihi kapasitas saluran air, terutama di selokan sungai [15].

2.1.1 Jenis Banjir

Terdapat berbagai macam banjir yang disebabkan oleh beberapa hal, diantaranya [15]:

1. **Banjir air**

   Banjir yang satu ini adalah banjir yang sudah umum. Penyebab banjir ini adalah meluapnya air sungai, danau, atau selokan sehingga air akan meluber lalu menggenangi daratan. Umumnya banjir seperti ini disebabkan oleh hujan yang turun terus-menerus sehingga sungai atau danau tidak mampu lagi menampung air.

2. **Banjir "Cileunang"**

   Jenis banjir yang satu ini hampir sama dengan banjir air. Namun banjir cileunang ini disebakan oleh hujan yang sangat deras dengan debit air yang sangat banyak. Banjir akhirnya terjadi karena air-air hujan yang melimpah ini tidak bisa segera mengalir melalui saluran atau selokan di sekitar rumah warga. Jika banjir air dapat terjadi dalam waktu yang cukup lama, maka banjir cileunang adalah banjir dadakan (langsung terjadi saat hujan tiba).

3. **Banjir bandang**

   Tidak hanya banjir dengan materi air, tetapi banjir yang satu ini juga mengangkut material air berupa lumpur. Banjir seperti ini jelas lebih berbahaya daripada banjir air karena seseorang tidak akan mampu berenang ditengah-tengah banjir seperti ini untuk menyelamatkan diri. Banjir bandang mampu menghanyutkan apapun, karena itu daya rusaknya sangat tinggi. Banjir ini biasa terjadi di area dekat pegunungan, dimana tanah pegunungan seolah longsor karena air hujan lalu ikut terbawa air ke daratan yang lebih rendah. Biasanya banjir bandang ini akan menghanyutkan sejumlah pohon-pohon hutan atau batu-batu berukuran besar. Material-material ini tentu dapat merusak pemukiman warga yang berada di wilayah sekitar pegunungan.

4. **Banjir rob (laut pasang)**

   Banjir rob adalah banjir yang disebabkan oleh pasangnya air laut. Banjir seperti ini kerap melanda kota Muara Baru di Jakarta. Air laut yang pasang ini umumnya akan menahan air sungan yang sudah menumpuk, akhirnya mampu menjebol tanggul dan menggenangi daratan.

5. **Banjir lahar dingin**

   Salah satu dari macam-macam banjir adalah banjir lahar dingin. Banjir jenis ini biasanya hanya terjadi ketika erupsi gunung berapi. Erupsi ini kemudian mengeluarkan lahar dingin dari puncak gunung dan mengalir ke daratan yang ada di bawahnya. Lahar dingin ini mengakibatkan pendangkalan sungai, sehingga air sungai akan mudah meluap dan dapat meluber ke pemukiman warga.

6. **Banjir lumpur**

   Banjir lumpur ini identik dengan peristiwa banjir Lapindo di daerah Sidoarjo. Banjir ini mirip banjir bandang, tetapi lebih disebabkan oleh keluarnya lumpur dari dalam bumi dan menggenangi daratan. Lumpur yang keluar dari dalam bumi bukan merupakan lumpur biasa, tetapi juga mengandung bahan dan gas kimia tertentu yang berbahaya. Sampai saat ini, peristiwa banjir lumpur panas di Sidoarjo belum dapat diatasi dengan baik, malah semakin banyak titik-titik semburan baru di sekitar titik semburan lumpur utama.

2.1.2 Penyebab Terjadinya Banjir

1. **Saluran Air yang Buruk**

   Pada kota-kota besar seperti Jakarta, Bandung, dan lainnya yang kerap terjadi biasanya dikarenakan saluran air yang mengalirkan air hujan dari jalan ke sungai sudah tidak terawat. Banyak saluran air di perkotaan yang tertutup sampah, memiliki ukuran yang kecil, bahkan tertutup beton bangunan sehingga fungsinya sebagai saluran air tidak dapat berjalan sebagaimana mestinya lalu kemudian terjadi genangan air di jalanan yang menyebabkan banjir.

2. **Daerah Resapan Air yang Kurang**

   Selain karena saluran air yang buruk ternyata daerah resapan air yang kurang juga mempengaruhi suatu wilayah dapat terjadi banjir. Daerah resapan air merupakan suatu daerah yang banyak ditanami pohon atau

yang memiliki danau yang berfungsi untuk menampung atau menyerap air ke dalam tanah dan disimpan sebagai cadangan air tanah.

Akan tetapi karena di daerah perkotaan seiring meningkatnya bangunan yang dibangun sehingga menggeser fungsi lahan hijau sebagai resapan air menjadi bangunan beton yang tentunya akan menghambat air untuk masuk ke dalam tanah. Sehingga terjadi genangan air yang selanjutnya terjadi banjir.

3. **Penebangan Pohon Secara Liar**

   Pohon memiliki fungsi untuk mempertahankan suatu kontur tanah untuk tetap pada posisinya sehingga tidak terjadi longsor, selain itu pohon juga memiliki fungsi untuk menyerap air sebagaimana telah disebutkan pada poin sebelumnya. Jika pada wilayah yang seharusnya memiliki pohon yang rimbun seperti daerah pegunungan ternyata pohonnya ditebangi secara liar, maka sudah pasti jika terjadi hujan pada daerah tersebut air hujannya tidak akan diserap ke dalam tanah tetapi akan langsung mengalir ke daerah rendah contohnya daerah hilir atau perkotaan dan perdesaan yang menyebabkan banjir.

4. **Sungai yang Tidak Terawat**

   Sungai sebagai media mengalirnya air yang tertampung dari hujan dan saluran air menuju ke laut lepas tentunya sangat memegang peranan penting pada terjadi atau tidaknya banjir di suatu daerah. Jika sungainya rusak dan tercemar tentu fungsinya sebagai aliran air menuju ke laut akan terganggu dan sudah dipastikan akan terjadi banjir.

   Biasanya kerusakan yang terjadi di sungai yaitu endapan tanah atau sedimentasi yang tinggi, sampah yang dibuang ke sungai sehingga terjadi pendangkalan, serta fungsi sempadan sungai atau bantaran sungai yang disalahgunakan menjadi pemukiman warga.

5. **Kesadaran Masyarakat yang Kurang Baik**

   Sikap masyarakat yang kurang sadar terhadap lingkungan juga ternyata sangat berpengaruh pada resiko terjadinya banjir. Sikap masyarakat yang kurang sadar mengenai membuang sampah agar pada tempatnya, menjaga

keasrian lingkungan, dan pentingnya menanami pohon menjadi faktor yang sangat penting untuk terjaganya lingkungan dan agar terhindar dari bencana banjir. Selain dapat menghindarkan banjir, sikap peduli lingkungan juga dapat menyehatkan dan tentunya akan meningkatkan taraf hidup masyaraktnya.

Dari kelima faktor di atas memang nampaknya kesadaran dari masyarakat untuk menjaga lingkungan sekitar sangat penting agar dapat terhindar dari banjir. Sangat percuma atau bahkan sia-sia jika program pemerintah dalam menanggulangi banjir seperti membangun kanal banjir, memugar saluran air, mengeruk sungai dari sedimentasi, dan yang lainnya jik atidak didukung oleh kesadaran warganya terhadap menjaga lingkungan [16].

2.1.3 Dampak yang Ditimbulkan Oleh Banjir

1. **Primer**

   Kerusakan fisik - Mampu merusak berbagai jenis struktur, termasuk jembatan, mobil, bangunan, sistem selokan bawah tanah, jalan raya, dankanal.

2. **Sekunder**

   Persediaan air – Kontaminasi air. Air minum bersih mulai langka.

   Penyakit - Kondisi tidak higienis. Penyebaran penyakit bawaan air.

   Pertanian dan persediaan makanan - Kelangkaan hasil tani disebabkan oleh kegagalan panen. Namun, dataran rendah dekat sungai bergantung kepada endapan sungai akibat banjir demi menambah mineral tanah setempat.

   Pepohonan - Spesies yang tidak sanggup akan mati karena tidak bisa bernapas.

   Transportasi - Jalur transportasi rusak, sulit mengirimkan bantuan darurat kepada orang-orang yang membutuhkan.

3. **Dampak tersier/jangka panjang**

   Ekonomi - Kesulitan ekonomi karena kerusakan pemukiman yang terjadi akibat banjir; dalam sector pariwisata, menurunnya minat wiasatawan;

biaya pembangunan kembali; kelangkaan makanan yang mendorong kenaikan harga, dll.

Dari berbagai dampak negatif yang ditimbulkan, ternyata banjir (banjir air skala kecil) juga dapat membawa banyak keuntungan, seperti mengisi kembali air tanah, menyuburkan serta memberikan nutrisi kepada tanah. Air banjir menyediakan air yang cukup di kawasan kering dan semi-kering yang curah hujannya tidak menentu sepanjang tahun. Air banjir tawar memainkan peran penting dalam menyeimbangkan ekosistem di koridor sungai dan merupakan faktor utama dalam penyeimbangan keragaman makhluk hidup di dataran. Banjir menambahkan banyak nutrisi untuk danau dan sungai yang semakin memajukan industri perikanan pada tahun-tahun mendatang, selain itu juga karena kecocokan dataran banjir untuk pengembangbiakan ikan (sedikit predasi dan banyak nutrisi) [15].

2.1.4 Penanggulangan Banjir

Mencegah dan menanggulangi banjir tak dapat dilakukan oleh pemerintah saja atau orang perorang saja. Dibutuhkan komitmen dan kerjasama berbagai pihak untuk menghindarkan Jakarta dan kota lain di Indonesia dari banjir besar. Tindakan-tindakan yang dapat dilakukan itu antara lain:

1. Membuang lubang-lubang serapan air
2. Memperbanyak ruang terbuka hijau
3. Mengubah perilaku masyarakat agar tidak lagi menjadikan sungai sebagai tempat sampah raksasa

Meninggikan bangunan rumah memang dapat menyelamatkan harta benda kita ketika banjir terjadi, namun kita tidak mencegah terjadinya banjir lagi. Manusia yang mengakibatkan banjir, manusia pula yang harus bersama-sama menyelamatkan kota. Menyelamatkan Jakarta dari banjir besar bukan hanya karena berarti menyelamatkan harta benda pribadi, namun juga menyelamatkan wajah bangsa ini di mata dunia.

Partisipasi seluruh elemen masyarakat harus dilakukan secara terorganisasi dan terkoordinasi agar dapat terlaksana secara efektif. Sebuah organisasi

masyarakat sebaiknya dibentuk untuk mengambil tindakan-tindakan awal dan mengatur peran serta masyarakat dalam penanggulangan banjir. Penanggulangan banjir dilakukan secara bertahap, dari pencegahan sebelum banjir penanganan saat banjir , dan pemulihan setelah banjir. Tahapan tersebut berada dalam suatu siklus kegiatan penanggulangan banjir yang berkesinambungan, Kegiatan penanggulangan banjir mengikuti suatu siklus (life cycle), yang dimulai dari banjir, kemudian mengkajinya sebagai masukan untuk pencegahan sebelum bencana banjir terjadi kembali. Pencegahan dilakukan secara menyeluruh, berupa kegiatan fisik seperti pembangunan pengendali banjir di wilayah sungai sampai wilayah dataran banjir dan kegiatan non-fisik seperti pengelolaan tata guna lahan sampai sistem peringatan dini bencana banjir [15].

2.2 Multimedia

Definisi multimedia secara umum adalah penggabungan berbagai informasi dengan menggunakan fasilitas dari komputer. Multimedia berasal dari kata multi yang berati banyak atau lebih dari satu dan media yang dapat diartikan penyajian suatu tempat. Multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, *audio*, gambar bergerak (video dan animasi) dengan menggunakan *link* dan *tool* yang memungkinkan pemakaian melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi. [10]

2.2.1 Objek Multimedia

Objek multimedia terdiri dari beberapa objek [1], yaitu:

1. Teks

   Teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia. Dalam kenyataannya multimedia menyajikan informasi kepada audiens dengan cepat, karena tidak diperlukan membaca secara rinci dan teliti. Sebagian besar sistem multimedia dirancang dengan menggunakan teks karena teks merupakan sarana yang efektif untuk mengemukakan ide-ide dan menyediakan instruksi-instruksi kepada *user* (pengguna).

2. Image

   Secara umum image atau grafik berarti still image seperti foto dan gambar. Manusia sangan berorientasi pada visual dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi.

3. Animasi

   Animasi adalah pembentukan gerakan dan berbagai media atau objek yang divariasikan dengan gerakan transisi, efek-efek, dan suara yang selaras dengan gerakan animasi tersebut atau animasi merupakan penayangan frame-frame gambar secara cepat untuk menghasilkan gerakan.

4. Audio

   Penyajian audio atau suara merupakan cara lain untuk lebih memperjelas pengertian suatu sistem informasi. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat melalui video. Suara dapat lebih menjelaskan karakteristik suatu gambar, misalnya musik dan suara efek (*sound effect*). Salah satu bentuk bunyi yang bisa digunakan dalam produksi multimedia adalah *Waveform Audio* yang merupakan format file audio yang berbentuk digital. Kualitas produknya bergantung pada *sampling rate* (banyaknya sampel per detik). *Waveform* (wav) merupakan standar untuk Windows PC.

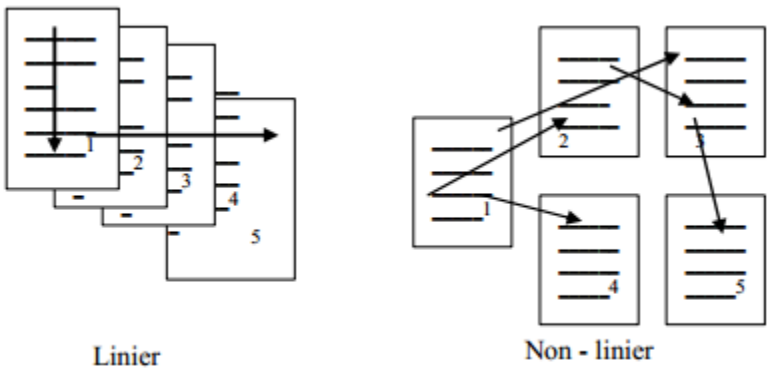
5. Video

   Video merupakan elemen multimedia paling kompleks karena penyampaian informasi yang lebih komunikatif dibandingkan gambar biasa. Walaupun terdiri dari elemen-elemen yang sama seperti grafik, suara, teks, namun bentuk video berbeda dengan animasi. Perbedaan terletak pada penyajiannya. Dalam video, informasi disajikan dalam kesatuan utuh dari objek yang dimodifikasikan sehingga terlihat saling mendukung penggambaran yang seakan terlihat hidup.

6. *Interactive link*

   Sebagian dari multimedia adalah interaktif, dimana pengguna dapat menekan *mouse* atau objek pada *screen* seperti *button* atau *teks* dan menyebabkan program melakukan perintah tertentu.

Interactive link dengan informasi yang dihubungkannya sering kali dihubungkan secara keseluruhan sebagai *hypermedia*. Secara spesifik, dalam hal ini termasuk *hypertext (hotward), hypergraphics* dan *hypersound*. *Interactive link* diperlukan untuk menggabungkan beberapa elemen multimedia sehingga menjadi informasi yang terpadu. Cara pengaksesan informasi pada multimedia terdapat dua macam, yaitu linier dan non-linier. Informasi linier adalah informasi yang ditampilkan secara sekuensial, yaitu dari atas ke bawah atau halaman demi halaman, sedangkan pada informasi non-linier (seperti gambar 2.1) dapat ditampilkan langsung sesuai dengan kehendak pengguna.



Gambar 2.1 *Interactive Link*

2.2.2 Kategori Multimedia

Multimedia dapat didefinisikan menjadi 2 kategori [1], yaitu:

1. Multimedia Linier

   Multimedia linier adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh pengguna. Contohnya: TV dan film.

2. Multimedia Interaktif

   Multimedia interaktif adalah suatu multimedia yang dilengkapi dengan alat pengontrol yang dapat dioperasikan oleh pengguna, sehingga pengguna dapat memilih apa yang dikehendaki untuk proses selanjutnya.

Contoh multimedia interaktif adalah multimedia pembelajaran interaktif, aplikasi game dan lain-lain.

2.3 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 [3].

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 kemajuab bermain dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.

2.3.1 Pengertian Game

Pengertian *game* menurut beberapa ahli [3]:

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 penggunaan 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) dan *fictitious* (pura-pura).
6. Menurut Greg Costikyan, *game* adalah sebentuk karya seni dimana peserta, yang disebut pemain membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan.

2.3.2 Sejarah Singkat Perkembangan Game

Dalam pengembangannya, *game* memiliki sejarah yang cukup panjang. Dari sebuah game yang sangat sederhana hingga game yang sangat kompleks. Dan dari sebuah komputer sederhana, hingga berbasis komputerisasi tinggi. Dari segi grafis rendah hingga *game* yang memiliki grafis tinggi. *Game* pertama yang dibuat adalah game *tic tac toe* oleh A. S. Douglas di tahun 1952. Dan game *tennis for two* yang dikembangkan oleh Willy Higginbotham. Pada tahun 1961, MIT membuat *game space war* menggunakan grafis vektor. Dan Sega merilis game *arcade* pertama pada tahun 1970. *Game* komersial pertama adalah *game computer space* yang merupakan pengembangan dari *game space war*. *Game* tersebut dikomersialkan oleh Nolan Bushnell pada tahun 1971. Pada tahun 1990, *game* terkenal dan merupakan *game platform adventure* legendaris Super Mario Bross dikembangkan dan diproduksi oleh Nintendo. *Game* tersebut merupakan *game*

dengan penjualan terbesar sepanjang sejarah. Dan hingga kini game semakin bervariatif dari segi *game play* dan grafis.

2.3.3 Jenis (*Genre*) Game

Game dapat dibagi ke beberapa genre diantaranya [6]:

1. Sport

   Genre yang satu ini terdiri dari game yang menciptakan tiruan dari *sports* asli. Jadi pemain bisa merasakan seperti bermain *sports* asli hanya dengan memainkan *game*-nya. Contohnya: *Pro Evolution Soccer, Fiva, Snow Board Hero*.



Gambar 2.2 Contoh dari permainan ber-*genre sports*

2. *Action*

   Genre ini menyediakan pemain dengan berbagai rintangan dan musuh yang harus dikalahkan dengan menggunakan reaksi cepat sang pemain dan biasanya tidak memberikan banyak teka-teki pada pemain. Contohnya: *Devil May Cry, Resident Evil*.



Gambar 2.3 Contoh permainan *Action*

3. *Adventure*

   Genre game ini terdiri dari *game* yang ada beberapa tantangan untuk pemain selama pemain mengendalikan 1 atau lebih karakter yang berkelana di tempat yang disediakan. Contohnya: *Space Quest IV, Labyrinth of Word, Tintin*.



Gambar 2.4 Contoh dari *game Adventure*

4. *Simulation*

   Genre game ini menyediakan simulasi dari aktifitas asli, jadi pemain diharapkan untuk mempunyai pengalaman yang sama tanpa melibatkan aktifitas asli. Contohnya: *Sim City, Roller Coster Tycoon, Caesar*.



Gambar 2.5 Contoh dari *Simulation Game*

5. *Strategy*

   Jenis permainan ini pemain diberi kesempatan untuk memikirkan strategi untuk menyelesaikan masalah atau mendapatkan gol, biasanya *setting*-nya tentang peperangan. Contohnya: *The Ancient Art of War, Command and Conquer* dan *Warcraft.*



Gambar 2.6 Contoh dari *Game Strategy*

6. *Puzzle*

   Game ini tentang pemain mencoba untuk menyelesaikan teka-teki yang diberikan oleh *game*. Contohnya: *Tetris*.



Gambar 2.7 Contoh dari *Game Puzzle*

7. *Role Playing Game (RPG)*

   Jenis permainan ini pemain diharapkan untuk mengikuti cerita dan berikan 1 karakter atau lebih untuk dikendalikan, karakter akan diberikan perjalanan dan akan mendapatkan pengalaman dengan mengalahkan musuh. Contohnya: *Final Fantasy series*, *Fire Emblem*.



Gambar 2.8 Contoh dari *Role Playing Game*

2.3.4 Game Edukasi

Game edukasi adalah *game* digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif.

Kriteria *game* edukasi yang ideal:

1. Rasa ingin tahu, fantasi dan kontrol pengguna.
    a. Motivasi intrinsik, menyenangkan untuk digunakan.
    b. Integrasi antara materi edukasi dan aspek fantasi permainan.
    c. Pengguna memegang kendali permainan dan dapat mengatur kecepatan bermain.
    d. Mendorong keingin tahuan, pengguna dapat melakukan eksplorasi bebas. Permainan mengandung rahasia tersembunyi.
    e. Pengguna dapat menyimpan kemajuan permainan.
    f. Menyediakan lebih dari satu jalur. Ada beberapa cara untuk memenangkan permainan.
    g. Keberhasilan ditentukan oleh pengetahuan, bukan kebetulan.
    h. Task disajikan secara incremental dan bertahap.
    i. Simulasi realistik dunia.
    j. Materi edukasi disesuaikan dengan materi dunia nyata.
    k. Personalisasi: pengguna dapat mengubah karakter dan objek-objek.
    l. Menyediakan banyak pilihan dan tema.

m. Hindari pengulangan, hindari *drill-practice*.
n. Sediakan elemen tidak terduga dan kejutan.

2. Tantangan
   a. Kinerja pemain terukur dan didefinisikan dengan baik.
   b. Tantangan diberikan secara terus-menerus tapi disesuaikan dengan tingkat pemahaman pemain.
   c. Kompleksitas semakin berkembang sejalan dengan kemampuan pengguna. Tersedia banyak level.
   d. Memonitor kinerja pengguna, berdasarkan hasil monitoring ini kesulitan kemudian disesuaikan.
   e. Pengguna dapat melihat progresnya setiap saat.
   f. Menyediakan hint dan instruksi untuk membantu pengguna.
3. Sosialisasi
   a. Pemain dapat berkolaborasi bersama.
   b. Pemain dapat saling berkompetisi.
   c. Pemenang bisa lebih dari satu (*multiple winner*).
4. Pedagogi
   a. Menyebutkan secara eksplisit target umur dan menyesuaikan rancangan permainan sesuai umur tersebut.
   b. Aktivitas pembelajaran dilaksanakan bersamaan dengan permainan.
   c. Objektif pembelajaran jelas.
   d. Memberi kesempatan kepada guru dan orangtua untuk ikut berperan.
   e. Memberikan petunjuk yang jelas sehingga pemain dapat berkonsentrasi kepada isi permainan, bukan bagaimana cara menggunakan permainan.
   f. Menyediakan tutorial sehingga pemain tidak perlu membaca manual.

5. Teknologi
   a. Menggunakan teknologi yang tersedia di sekolah dan masyarakat umum (tidak membutuhkan hardware yang terlalu tinggi).
   b. Memberikan lisensi terjangkau sekolah.
   c. Menggunakan antarmuka yang intuitif.

6. Pengguna anak-anak dan yang berkebutuhan khusus
    a. Memberikan petunjuk yang dibacakan selain petunjuk tertulis.
    b. Gambar, objek, layar tertata rapi.
    c. Permainannya sendiri merupakan aktivitas yang penting (*play for the sake of play*).
    d. Transformasi terlihat. Saat anak melakukan interaksi, ada instant *feedback*.
    e. Input dan output yang mudah terlihat.
    f. Tantangan diberikan secara gradual.
    g. Permainan menyenangkan untuk diulang, beberapa karakter, musik mudah teringat oleh anak.
    h. Objek dan suara diambil dari kehidupan sehari-hari yang mudah dikenali.
    i. Menginspirasi anak, bahkan setelah komputer dimatikan.

2.3.5 Game *Design*

Game design merupakan proses menciptakan konten dan aturan permainan. *Design game* yang baik adalah proses menciptakan tujuan dimana seorang pemain merasa termotivasi untuk mencapainya dan seorang pemain harus mengikuti aturan saat dia membuat keputusan yang berarti dalam mengejar tujuan-tujuan tersebut. Menurut Brenda Brathwaite dan Ian Schreiber dalam bukunya *Challanges For Game Designers*, merancang *game* mencakup sebagai berikut [2]:

a. *World Design*, membuat desain keseluruhan latarbelakang cerita, pengaturan, dan tema permainan.
b. *System Design*, membuat desain aturan dan pola matematis yang mendasari dalam sebuah permainan memiliki aturan.
c. *Content Design*, membuat desain karakter, item, teka-teki dan misi.
d. *Game Writing*, membuat penulisan dialog, text dan cerita dalam dunia permainan.

e. *Level Design*, menciptakan level dalam permainan berdasarkan tampilan permainan dan penempatan objek hingga tantangan dalam tampilan permainan.
f. *User Interface*, membuat desain yang terdiri dari dua hal, yaitu bagaimana pemain dapat berinteraksi dengan permainan dan bagaimana pemain menerima informasi dan mengambil hasil dari permainan.

Para pengembang *game* komersial biasanya mempersiapkan dokumen-dokumen yang berisi *design game* yang sangat panjang sebelum memulai pembuatan *game*. Setelah itu dari *design game* yang telah dibuat kemudian dapat diketahui semua elemen-elemen berbeda yang dibutuhkan dalam pembuatan *game*, misalnya karakter user, karakter musuh, animasi serangan dan sebagainya. Membuat *game* akan membutuhkan gambar dari tiap elemen-elemen yang ada, *background image* dan lagu. Semua hal tersebut dapat dikatakan sebagai *resource game*. Hal-hal yang penting dalam *game* adalah sebagai berikut [18]:

1. *Sprites*

   Sprites adalah kumpulan gambar-gambar yang digunakan untuk icon, karakter ataupun benda bergerak dalam *game*. *Sprites* dapat dibuat dari gambar-gambar yang sudah dibuat dalam sebuah *art packages*, download dari internet maupun dibuat secara manual dengan menggunakan perangkat lunak seperti *Adobe Photoshop*.
2. *Object*

   Sprites tidak dapat melakukan sesuatu dengan sendirinya. *Sprites* hanya menyimpan gambar-gambar dari elemen yang berbeda dalam sebuah fame. *Object* adalah bagian dari *game* yang mengontrol bagaimana elemen-elemen tersebut bergerak dan berinteraksi satu sama lain. Contohnya adalah karakter dalam *game*.
3. *Events* dan *Actions*

   Events adalah hal-hal penting yang terjadi didalam sebuah *game*, seperti ketika *objects* saling bertubrukan maupun ketika user menekan tombol pada *keyboard*. *Actions* adalah hal-hal yang terjadi sebagai respon dari

sebuah *event*, sepeti perubahan arah perbaikan objects, perubahan *score* atau memainkan sebuah lagu.

4. *Backgrounds*

   Backgrounds adalah gambar-gambar yang digunakan sebagai gambar latar belakang *game* dan merupakan jenis lain dari *resource*, seperti *sprites* dan *objects*. Pada umumnya ukuran *backgrounds* sama dengan resolusi *game*.

5. *Sounds*

   Sounds jg termasuk *resource game*, jenis *sounds* dalam *game* ada dua, yaitu *music* dan *sound effect*. *Sounds* dapat mempengaruhi atmosfer dalam *game*. *Masters* berupa lagu yang terus mengalun selama *game* dimainkan, namum lagu yang dimainkan pada umumnya bergantung pada tempat atau kondisi *game*. *Sounds effect* adalah suara yang hanya terdengar ketika terjadi suatu hal tertentu, misalkan penekanan tombol, karakter terkena pukulan dan sebagainya. Kegunaan dari *sound effect* adalah untuk membantu user mengetahui *actionaction* yang telah dilakukan maupun yang sedang terjadi.

6. Status

   Status adalah sekumpulan informasi yang berguna untuk orang yang memainkan sebuah *game* untuk mengetahui kondisi permainan saat itu. Bergantung dari jenis *game*-nya, informasi yang disampaikan melalui status dapat bermacam-macam. Berikut beberapa macam status pada *game* beserta penjelasannya:

   a. *Score* adalah nilai yang diperoleh oleh orang yang memainkan sebuah *game*. Status ini sangat sering digunakan pada sebuah *game* untuk mengetahui kemampuan yang memainkannya.
   b. *High Score* adalah nilai tertinggi yang pernah dicapai orang yang memainkan sebuah *game*. Status ini biasanya terdapat pada *game* yang dapat diselesaikan dalam waktu yang cukup singkat, sehingga membuat orang yang memainkannya tertarik untuk mengulangi permainan dari awal, dengan harapan meningkatkan *high scorehigh score* yang lebih tinggi.

c. HP (*Hit Point*) adalah status yang menunjukkan jumlah maksimal *damage* yang dapat diterima oleh karakter dalam *game*. Apabila bernilai 0 maka karakter tersebut dinyatakan mati/kalah. Status ini biasany terdapat pada *game* ber-*genre fighting, action* atau RPG (*Role Playing Game*).

d. SP (*Skill Point*) adalah sejumlah poin yang dapat digunakan untuk menggunakan *skill*. Jumlah yang dibutuhkan bergantung dari *skill* yang digunakan. Status ini biasany terdapat pada *game* ber-*genre* RPG (*Role Playing Game*).

e. *Strength* adalah sejumlah kekuatan karakter dalam *game*. Nilai status biasanya menentukan kekuatan serangan karakter. Status ini biasanya terdapat pada *game* ber-*genre* RPG (*Role Playing Game*).

f. *Vitality* adalah jumlah pertahanan karakter dalam *game*. Nilai status ini biasanya akan mempengaruhi jumlah HP dan pertahanan yang dimiliki karakter. Status ini biasanya terdapat pada *game* ber-*genre fighting, action* atau RPG (*Role Playing Game*).

g. *Speed* adalah kecepatan karakter dalam *game*. Nilai status ini biasanya akan mempengaruhi seberapa cepat karakter akan bergerak atau mendapatkan giliran serta menentukan mudah atau tidaknya tekena serangan musuh. Status ini biasanya terdapat pada *game* ber-*genre fighting, action* atau RPG (*Role Playing Game*).

h. *Intelligence* adalah jumlah kekuatan sihir dari karakter dalam *game*. Nilai status ini biasanya akan berpengaruh pada jumlah SP, serangan sihir dan ketahanan terhadap serangan sihir. Status ini biasanya terdapat pada *game* ber-*genre fighting* atau RPG (*Role Playing Game*).

i. *Luck* adalah keberuntungan dalam *game*. Nilai status ini biasanya akan mempengaruhi karakter dalam mendapatkan hal-hal yang sulit didapatkan dalam *game*, seperti terjadinya *critical hit*.

j. Status *point* adalah sejumlah poin yang biasany digunakan untuk menambah status karakter seperti *Strength, Vitality, Intelligence,*

Speed dan *Luck*. Status ini biasanya terdapat pada *game* ber-*genre action* atau RPG (*Role Playing Game*).

k. *Time* adalah status yang digunakan untuk berbagai macam fungsi dalam *game*, misal untuk menentukan waktu yang tersisa untuk mencapai tujuan tertentu, jumlah waktu yang telah dihabiskan seseorang untuk memainkan *game* tersebut dan lain sebagainya. Status ini biasanya terdapat pada *game* ber-*genre fighting, action* atau RPG (*Role Playing Game*).

l. *Experience* dalam dunia *game* adalah sejumlah poin yang biasanya digunakan untuk menentukan *level* karakter dalam *game*. Status ini biasanya terdapat pada *game* ber-*genre action* atau RPG (*Role Playing Game*).

2.3.6 Metode Animasi

Animasi merupakan suatu teknik menampilkan gambar berurut sedemikian rupa sehingga penonton merasakan adanya ilusi gerakan (motion) pada gambar yang ditampilkan [13]. Animasi dapat dibuat dengan tiga teknik berbeda, yaitu *Image*, *Xoring* dan *Make*. Pergerakannya dapat bertipe *object sprite* atau *object frame*. Metode animasi yang digunakan antara lain animasi *frame, bibliting* dan *realtime*. Walaupun terbagi atas berbagai definisi berbeda, tapi dalam prakteknya teori-teori tersebut dapat digabungkan atau saling berhubungan sehingga tidak murni dipakai sendiri. Macam-macam animasi yang digunakan dalam membuat sebuah *game* akan dijelaskan sebagai berikut [12]:

a. Animasi dengan Teknik *Image*

   Animasi dengan teknik ini adalah menyimpan *image* sebagai sebuah *sprite* dalam memori yang kemudian akan ditampilkan di *background*-nya. Dalam teknik ini animasi yang disimpan harus berlatar belakang sesuai background-nya. Animasi dengan teknik ini biasanya sulit dalam pembuatan gambarnya, sebab harus banyak dan melakukan penyamaan posisi. Akan tetapi teknik ini mudah dalam hal memainkan animasinya.

b. Animasi dengan Teknik *Xoring*

Teknik ini adalah teknik animasi yang mudah dan sederhana, sebab selain gambarnya satu *sprite*, cara menampilkannya juga jauh lebih mudah dibanding dengan teknik sebelumnya. Pembuatan gambarnya sangat mudah, sebab yang dibuat adalah *sprite*-nya saja dan tidak perlu menyamakan dengan *background*-nya. Kelemahan dari teknik ini adalah memiliki efek buruk yaitu tembus pandang dan mengganti warna *sprite*, maka hal ini tidak baik digunakan dalam animasi yang *background*-nya bergambar.

c. Animasi dengan Teknik *Make*

Animasi dengan teknik ini biasanya digunakan untuk animasi umum, tapi biasanya digunakan untuk proses pembuatan animasi. Animasi dengan teknik ini memiliki *sprite* yang terus-menerus digenerate oleh program, kemudian ditampilkan dengan perhitungan tertentu. Animasi ini biasanya dilakukan oleh 3D *modelling* dan *shading* oleh *software* seperti AutoCad, 3D Studio, Presidio 3D Workshop dan lain-lain. Operasi diatas haruslah di generate secara langsung dengan perhitungan sehingga pembuatan hampir bersamaan dengan saat menampilkannya.

d. Animasi dengan Tipe *Object Sprite*

Animasi ini menggunakan *sprite* sebagai pemeran utama sedangkan *object* lainnya hanya *background* diam. Prosesnya adalah membuat gambar *sprite* dengan latar belakang warna hitam, lalu dibuat juga *sprite* yang sama tetapi berwarna hitam dan latar belakangnya adalah warna tertinggi, kemudian ditempatkan dengan pertama-tama menyimpan *background* yang akan ditimpa oleh *sprite* dan ditempatkan *sprite* dengan Xor dan ditimpa dengan warna tertinggi secara Xor.

e. Animasi dengan Tipe *Object Frame*

Animasi ini menitik beratkan animasi yang dimainkan hanya pada *sprite object*-nya saja, akan tetapi keseluruhan *background*-nya juga seolah-olah ikut digerakkan.

f. Metode Animasi *Frame*

Metode ini adalah metode animasi yang mendukung tipe *object frame*. Karena metode animasi dengan metode *fullscreen*, maka *frame* yang tampil haruslah disiapkan terlebih dahulu dalam beberapa page sebelumnya. Karena hal tersebut pengambilan gambarnya haruslah sangat cepat, sehingga tidak menjadikan animasi lamban dan tersendat. Animasi *frame* ini haruslah menampilkan gambar *fullscreen* yang bergerak, agar efek tersendat dan pergantian *frame* tidak menyolok.

g. Metedo Animasi *BitBlt*

Metode animasi ini biasanya disebut *sprite animation, array animation, blocked animation, partial screen animation, snapshot animation* atau *arcade animation*. Prinsip dari metode ini adalah menyimpan *image* dan memainkan animasinya dalam bentuk satu atau beberapa *sprite* kecil. BitBlt ini digunakan untuk membuat program permainan dan program berbasis grafis terutama dalam membuat animasi sederhana.

h. Metode Animasi *Real-Time*

Dalam metode ini biasanya semua animasi yang sedang tampil atau yang akan dibuat dilakukan bersama sehingga tidak perlu disiapkan terlebih dahulu. Karena animasi ini lambat dan tersendat maka animasi dengan metode ini akan bagus jika pergerakan yang akan dilakukan adalah tidak diketahui sebelumnya dan tiba-tiba muncul. Metode ini tidak disarankan untuk animasi biasa-biasa saja, akan tetapi sebaiknya digunakan untuk keperluan khusus seperti *rotating* dan *tweening*.

2.4 Artificial Intelligence

Artificial Intelligence adalah ilmu yang berusaha membangun entitas-entitas cerdas yang sesuai dengan pemahaman manusia. Entitas-entitas cerdas yang dibangun AI ternyata sangat menarik dan mempercepat proses pemahaman terhadap kecerdasan manusia. Hingga saat ini, AI terus dipelajari dan dikembangkan secara meluas maupun mendalam. Pada saat ini, banyak bidang studi baru yang berawal dari AI, seperti *computational intelligence, soft*

computing, fuzzy system, evolutionary computation dan banyak lagi lainnya yang semakin fokus pada bidang kajian dan permasalahan tertentu [11].

2.4.1 Pengertian Artificial Intelligence

Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Stuart Russel dan Peter Norvig mengelompokan empat definisi AI yaitu [5]:

1. *Thingking Humanly : the cognitive modeling approach.*

   Pendekatan ini dilakukan dengan dua cara, yang pertama yaitu dengan melalui instropeksi, dimana dengan cara tersebut mencoba menangkap pemikiran-pemikiran kita sendiri pada saat kita berpikir. Dan yang kedua adalah dengan cara bereksperimen psikologi, dimana dengan cara tersebut kita mempelajari penghayatan emosi kita sendiri.

2. *Acting Humanly : the turing test approach.*

   Pada tahun 1950, Alan Turing merancang suatu ujian bagi komputer berintelejensia untuk menguji apakah komputer tersebut mampu mengelabui seorang manusia yang mengintrogasinya melalui teletype atau komunikasi berbasis teks jarak jauh. Jika interrogrator tidak dapat membedakan yang diinterogasi adalah manusia atau komputer, maka komputer berintelejensia tersebut lolos dari Turing test. Komputer tersebut perlu memiliki kemampuan: *Natural Language Processing, Knowledge Representation, Automated Reasoning, Machine Learning, Computer Vision, Robotics. Turing test* sengaja menghindari interaksi fisik antara *interrogator* dan komputer karena simulasi fisik manusia tidak perlu intelejensia.

3. *Thingking Rationaly : the laws of thought approach.*

   Terdapat dua masalah dalam pendekatan ini, yaitu:

   1. Tidak mudah untuk membuat pengetahuan informal dan menyatakan pengetahuan tersebut ke dalam *formal term* yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastian kurang dari 100%.

2. Terdapat perbedaan besar antara dapat memecahkan masalah "dalam prinsip" dan memecahkannya "dalam dunia nyata".

4. *Acting Rationaly : the rational agent approach.*

   Membuat inferensi yang logis merupakan bagian dari suatu *rational agent*. Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Dengan menalar secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akan mencapai tujuan atau tidak. Jika mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut.

2.4.2 Perbandingan Kecerdasan Buatan dengan Kecerdasan Alami

Ada beberapa keuntungan kecerdasan buatan dengan kecerdasan alamiah diantaranya adalah:

1. Kecerdasan buatan lebih bersifat permanen tidak seperti kecerdasan alami yang akan cepat mengalami perubahan karena sifat manusia yang pelupa.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarkan tidak seperti manusia karena mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang lama dan juga suatu keahlian itu tidak akan pernah dapat diduplikasikan dengan lengkap.
3. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
4. Kecerdasan buatan bersifat konsisten.
5. Kecerdasan buatan dapat didokumentasikan.
6. Kecerdasan buatan akan lebih cepat mengerjakan pekerjaan dibanding dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami [4].

2.4.3 Algoritma Greedy

Algoritma Greedy merupakan algoritma yang paling sering digunakan untuk memecahkan masalah optimasi. Greedy berarti rakus atau tamak, hal tersebut berarti algroritma greedy mengambil solusi paling tinggi untuk setiap

langkahnya. Prinsip dari algoritma greedy adalah *Take What You Can Get Now* [11].

Pada algoritma greedy setiap langkahnya tidak dapat diulang kembali, oleh karena itu setiap langkah dari algoritma greedy diputuskan dengan nilai solusi yang paling optimum. Pendekatan dari algoritma greedy yaitu dengan memberikan pilihan terbaik dengan membuat pilihan optimum lokal dimana setiap langkah mengarah ke solusi optimum. Jadi jika pada langkah optimum lokal berhenti, diharapkan solusi tersebut menjadi solusi optimum.

Algoritma greedy disusun oleh elemen-elemen berikut [11]:

1. Himpunan Kandidat

   Berisi elemen-elemen pembentuk solusi.
2. Himpunan Solusi

   Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi Seleksi

   Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi Kelayakan

   Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi Objektif

   Fungsi yang memaksimumkan atau meminimumkan nilai solusi.

2.4.4 *Collision Detection*

Dalam sebuah game membutuhkan respon dari masing-masing objek disaat saling bersentuhan. Respon sentuhan tersebut disebut dengan collision detection. Terdapat beberapa jenis atau metode dari collision detection tersebut

diantaranya adalah metode *bounding circle, test, bounding box test* dan *line intersection test*.

Bounding box test merupakan metode collision dimana setiap objek memiliki garis tepi yang akan mendeteksi apakah objek tersebut bertumbukan dengan objek lain atau tidak. Garis tepi tersebut didefinisikan dari garis tepi kiri, garis tepi bawah, garis tepi kanan dan garis tepi atas [14].

2.5 Pemrograman Berorientasi Objek

Metodologi berorientasi objek aalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data an operasi yang diberlakukan terhaapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktivitas karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*)
2. Kecepatan pengembangan karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan [7].

2.5.1 Karakteristik Metodologi Pemrograman Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama [7], yaitu:

1. *Encapsulation*

   Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindungi dari prosedur atau objek lain kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance*

   Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi atribut dan metoda dari induknya langsung. Atribut dan metoda dari objek induk diturunkan kepada anak objek, demikian seterusnya. Pendefinisian objek dipergunakan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak perlu membuat atribut dan metoda lagi pada anaknya, karena telah mewarisi sifat induknya.

3. *Polymorphism*

   Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

2.6 Unified Modeling Language (UML)

UML atau *Unified Modeling Language*, merupakan pemodelan berorientasi objek yang dikembangkan oleh Grady Bosch dan James Rumbaugh

dari Rational Software Corporation pada akhir tahun 1994 dengan menggabungkan metode Booch dan OMT (*Object Modeling Technique*). Lalu pada tahun 1995, Ivar Jacobson dan perusahaan Objectory bergabung dengan Rational lalu menggabungkan metode OOSE (*Object Oriented Software Engineering*) dan dari ketiga penggabungan tersebut lahirlah UML sebagai metode baru.

Berikut merupakan beberapa diagram UML diantaranya [8]:

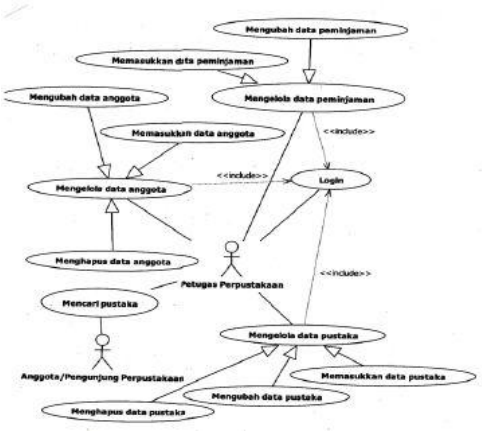
A. *Use Case Diagram*

Use Case atau diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior*) dari sistem yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

   Contoh dari *use case diagram* dapat dilihat pada gambar 2.9.



Gambar 2.9 Contoh *Use Case Diagram*

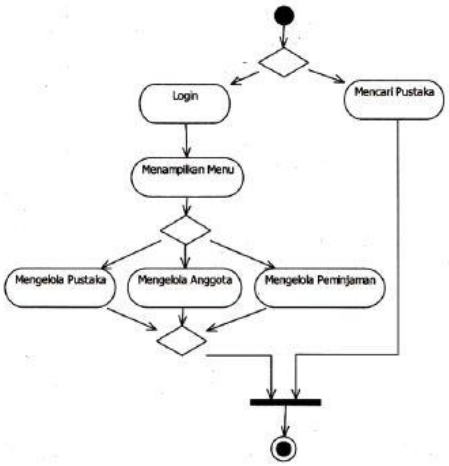
B. *Activity Diagram*

Diagram aktivitas atau *activity diagram* adalah sebuah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Dalam diagram aktivitas yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-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/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

   Contoh dari *activity diagram* dapat dilihat pada gambar 2.10.



Gambar 2.10 Contoh dari *Activity Diagram*

C. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

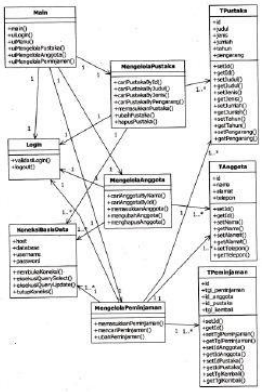
1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Pendefinisian kelas terdiri dari:

1. Boundary atau Kelas pembatas merupakan class yang menyalurkan interaksi antara sistem dengan dunia sekitarnya. Seperti form, laporan, obyek–obyek pada form. Setiap kelas pembatas biasanya akan mewakili interaksi antara seorang actor dengan use case.
2. Entity atau kelas entitas biasanya digunakan untuk menangani informasi yang mungkin akan selalu disimpan dalam proses bisnis. Cara melakukan identifikasi kelas entitas adalah dengan memperhatikan kata b*end*a. Seperti kalimat "petugas mencatat peminjaman buku", buku bisa menjadi *Class* entitas. Kelas entitas juga dapat digunakan untuk mewakili table –table yang terdapat dalam database.
3. Control atau kelas control bersifat opsional, apabila kelas ini digunakan maka satu kelas *control* untuk satu *use case* yang digunakan mengatur kejadian dalam use case tersebut. Contohnya kelas transaksi yang

bertanggung jawab dalam transaksi baik peminjaman maupun pengembalian buku dalam suatu perpustakaan.

Contoh dari *class diargram* dapat dilihat pada gambar 2.11.



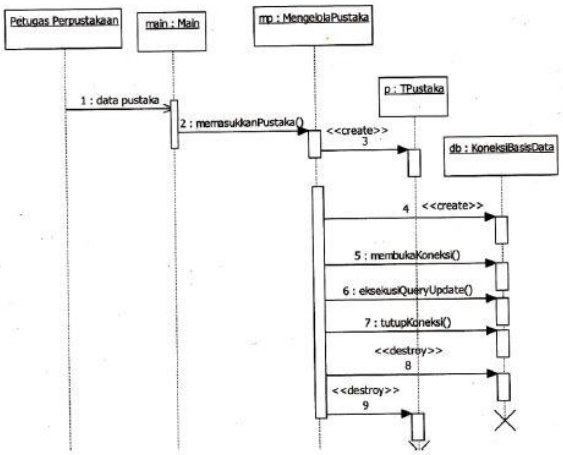
Gambar 2.11 Contoh dari *Class Diagram*

D. *Sequence Diagram*

Diagram sekuen adalah diagram yang menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyak diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses itu sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Contoh dari *sequence diagram* dapat dilihat pada gambar 2.12.



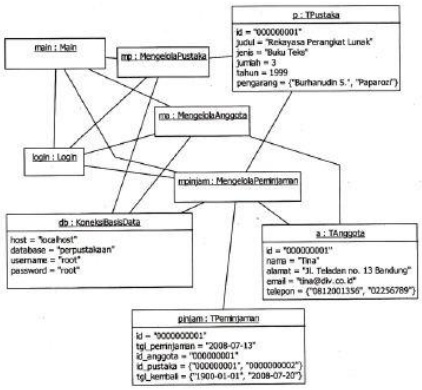
Gambar 2.12 Contoh dari *Sequence Diagram*

E. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalanny objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai. Hubungan *link* pada diagram objek merupakan hubungan memakai dan dipakau dimana dua buah objek akan dihubungkan oleh *link* jika ada objek yang dipakai oleh objek lainnya.

Contoh dari *object diagram* dapat dilihat pada gambar 2.13.



Gambar 2.13 Contoh dari *Object Diagram*

F. *Component Diagram*

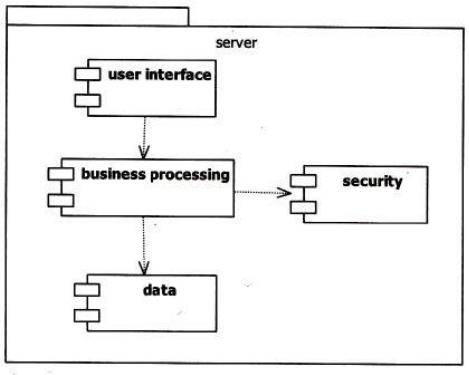
Diagram komponen dibuat untuk menunjukkan organisasi dan kebergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

1. *Source code* program perangkat lunak
2. Komponen *executable* yang dilepas ke *user*
3. Basis data secara fisik
4. Sistem yang harus beradaptasi dengan sistem lain
5. *Framework* sistem

Adapun komponen-komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut:

1. Komponen *user interface* yang menangani tampilan
2. Komponen *business processing* yang menangani fungsi-fungsi proses bisnis
3. Komponen *data* yang menangani manipulasi data
4. Komponen *security* yang menangani keamanan sistem

Contoh dari *componen diagram* dapat dilihat pada gambar 2.14



Gambar 2.14 Contoh dari *Component Diagram*

G. *Composite Structure Diagram*

Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung. Contoh penggunaan diagram ini misalnya menggambarkan dekripsi dari setiap bagian mesin yang saling terkait *router* pada jaringan komputer dan lain-lain.

H. *Package Diagram*

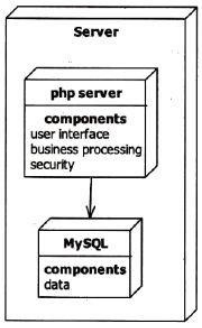
Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML.

I. *Deployment Diagram*

Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan device, node dan selanjutnya
2. Sistem *client/server*
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

   Contoh dari *deployment diagram* dapat dilihat pada gambar 2.15

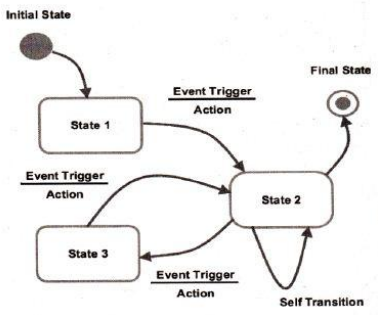


Gambar 2.15 Contoh dari *Deployment Diagram*

J. *State Mchine Diagram*

Diagram mesin digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu graf berarah. *State machine diagram* merupakan pengembangan dari diagram *Finite State Automata* dengan penambahan beberapa fitur dan konsep baru.

Diagram ini cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem. Contoh dari *state machine diagram* dapat dilihat pada gambar 2.16.



Gambar 2.16 Contoh dari *State Machine Diagram*

K. *Communication Diagram*

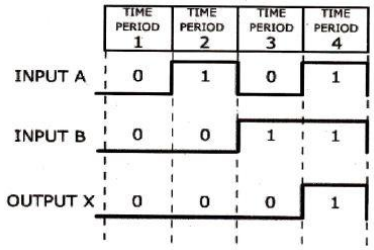
Diagram komunikasi merupakan penyederhanaan dari diagram kolaborasi (*collaboration diagram*). Diagram ini menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi mempresentasikan informasi yang diperoleh dari Diagram Kelas, Diagram Sekuen dan Diagram *Use Case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

Diagram komunikasi mengelompokkan *message* pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi/metode yang dijalankan antara objek yang satu dan objek yang lainnya secara keseluruhan. Oleh karena itu dapat diambil dari jalannya interaksi pada semua diagram sekuen. Penomoran metode dapat dilakukan berdasarkan urutan dijalankannya metode/operasi diantara objek yang satu dengan objek lainnya atau objek itu sendiri.

L. *Timing Diagram*

Timing Diagrami merupakan diagram yang fokus pada penggambaran terkait batasan waktu. *Timing Diagram* digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu. *Timing Diagram* biasanya digunakan untuk mendeskripsikan operasi dari alat *digital* karena penggambaran secara visual akan lebih mudah dipahami daripada dengan kata-kata. Aliran waktu pada *timing diagram* dibaca dari kiri ke kanan.

Contoh dari *timing diagram* dapat dilihat pada gambar 2.17.



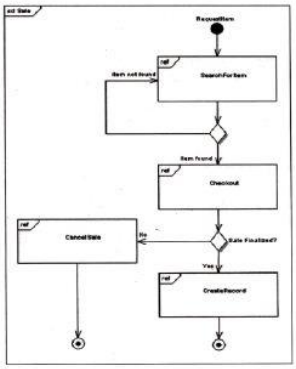
Gambar 2.17 Contoh dari *Timing Diagram*

M. *Iteraction Overview Diagram*

Iteraction overview diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. *Iteraction overview diagram* adalah bentuk aktivitas diagram yang setiap titik mempresentasikan diagram interaksi. Interaksi diagram dapat meliputi diagram sekuen, diagram komunikasi, *iteraction overview diagram* dan *timing diagram*.

Hampir semua notasi pada *iteraction overview diagram* sama dengan notasi pada diagram aktivitas. Sebagai contoh *initial, final, decision, merge, fork* dan *join nodes* sama seperti pada diagram aktivitas. Tambahan pada *iteraction overview diagram* adalah *interaction accurrence* dan *interaction element*.

Contoh dari *overview diagram* dapat dilihat pada gambar 2.18.



Gambar 2.18 Contoh dari *Iteraction Overview Diagram*

2.7 Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan untuk merancang dan membangun *game* Cegah Banjir ini sebagai berikut.

2.7.1 Game Maker

Game maker adalah *game engine* yang sangat populer untuk membuat sebuah *game*. Game maker memberikan resource yang cukup banyak, dari segi *grafis* 2D hingga 3D, *sprite maker, sound, script, path* dan sebagainya.

Game maker dibuat oleh Mark Overmars, seorang profesor dari Institute Ilmu Komputer dan Informasi di Universitas Utrecht, dengan pemrograman Delphi 7. Awalnya pada tahun 1999 Game Maker bersama Animo, sebuah program yang memiliki sistem pembuat animasi dua dimensi. Namun, pada tahun yang sama Animo telah berubah fungsi menjadi sebuah *game engine* tetapi tidak dapat membuat sebuah program *executable,* hanya bisa dimainkan pada jendela Game Maker. Seiring perkembangan teknologi, Game Maker dapat membuat game utuh dan mendukung DirectX.

Game Maker memiliki program terpisah untuk setiap pembuatan jenis game. Untuk pembuatan *game* berbasis web, Game Maker memberikan tools program Game Maker HTML5. Untuk pembuatan game berbasis PC (*Personal Computer*), tools programnya adalah Game Maker 8.0 untuk *platform* dengan sistem operasi windows dan mac. Game Maker pun mempunyai *engine* untuk *smartphone*, yaitu Game Maker Studio. Game Maker Studio adalah *game engine*

untuk pembuatan *game* di platform iOS, Android OS dan Windows Phone. Game Maker Language merupakan bahasa pemrograman untuk pengolahan script di Game Maker. Game Maker Language merupakan bahasa pemrograman berorientasi objek. Penggunaan Game Maker Language merepresentasikan setiap objek yang dibuat dan memberikan suatu perintah kepada objek tersebut. Game Maker Language dibuat oleh Mark Overmars, pembuat Game Maker sendiri.

2.7.2 GIMP 2.0

Gimp atau *GNU Image Manipulation Program*, merupakan perangkat lunak pengolahan grafis yang dikembangkan pertama kali oleh Peter Mattis. Awalnya Gimp merupakan kependekan dari *General Image Manipulation Program* yang berjalan di sistem operasi Linux, namun seiring perkembangan teknologi Gimp dikembangkan untuk setiap sistem operasi Windows. Gimp memiliki fitur yang sangat luas, dari pengedit foto atau pembuat gambar. File grafis yang dihasilkan oleh Gimp adalah berekstensi .xcf namun dapat mengekspor ke berbagai file grafis lainnya seperti .JPG, .PNG, .ICO ataupun .GIF. Gimp juga dapat mengimpor file grafis .PSD, yang merupakan file grafis yang dihasilkan oleh pengolahan grafis photoshop.

2.7.3 StarUML

StarUML merupakan *open source* alat UML, berlisensi versi modifikasi dari GNU GPL. Setelah ditinggalkan untuk beberapa waktu, proyek ini memiliki kebangkitan terakhir yang pindah dari Delphi untuk Java/Eclipse dan berhenti lagi. Namun masyarakat masih aktif dan banyak topik yang dibahas di forum.

Tujuan dari proyek ini adalah untuk menggantikan yang lebih besar, aplikasi komersial seperti Rational Rose dan Borland’s Together. StarUML mendukung sebagian besar jenis diagram ditentukan dalam UML 2.0. Saat ini tidak ada objek, paket, waktu dan diagram gambaran interaksi (meskipun dua yang pertama bisa cukup dimodelkan melalui diagram kelas editor).

StarUML ditulis dalam Delphi, yang merupakan salah satu alasan mengapa hal ini ditinggalkan untuk waktu yang lama. Dalam StarUML ada ekspor

ke gambar raster format JPEG dan vektor format gambar Windows Metafile. Tapi tidak ada ekspor ke SVG dan PNG. Untuk impor, bisa impor dari XMI dan Rational Rose format file.

2.8 Skala Likert

Skala Likert menurut Djaali (2008:28) ialah skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu gejala atau fenomena pendidikan. Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, pendidik dan ahli psikolog Amerika Serikat. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932.

Skala Likert itu "aslinya" untuk mengukur **kesetujuan dan ketidaksetujuan** seseorang terhadap sesuatu objek, yang jenjangnya bisa tersusun atas:

1. sangat setuju
2. setuju
3. sedang
4. tidak setuju
5. sangat tidak setuju.

Pernyataan yang diajukan mengenai objek penskalaan harus mengandung isi yang akan "dinilai" responden, apakah setuju atau tidak setuju. Apabila data yang diperoleh melalui jawaban kuesioner telah terkumpul, maka angka dalam bentuk skala likert yang dipilih responden (pada setiap pernyataan pada kuesioner) selanjutnya dikalkulasi. Perhitungan ini menggunakan teknik statistik sehingga dapat dianalisis lebih lanjut.