BAB 2
TINJAUAN PUSTAKA

2.1 Sejarah Komplek Pemerintahan Kabupaten Bandung

Kabupaten Bandung berdiri pada tanggal 20 April 1641 M. Pada saat itu Tumenggung Wiraangun-angun (1641-1681) sebagai Bupati Bandung Pertama, pusat pemerintahan Kabupaten Bandung saat itu berada didaerah "Krapyak", sebuah daerah yang tidak jauh dari aliran sungai Citarum diseputar kawasan Dayeuhkolot sekarang. Dalam istilah Belanda, Dayeuhkolot disebut dengan nama "Oude Negorij", atau Negara Lama. Pada masa sekarang daerah itu dinamakan Kampung Citeureup yang berada di wilayah Kecamatan Dayeuhkolot. Pada masa Pemerintahan Adipati Wiranatakusumah II (1794-1829) ibu kota Kabupaten Bandung dipindahkan dari Karapyak (Dayeuhkolot) ke tepi sungai Cikapundung atau alun-alun Kota Bandung sekarang, pemindahan ibu kota itu atas dasar perintah dari Gubernur Jenderal Hindia Belanda Daendels tanggal 25 Mei 1810, dengan alasan daerah baru tersebut dinilai akan memberikan prospek yang lebih baik terhadap perkembangan wilayah tersebut. Kemudian pada masa transisi (Orde Lama ke Orde Baru) pada masa pimpinan Kolonel R.H. Lily Sumantri tercatat peristiwa penting yaitu rencana pemindahan ibu kota Kabupaten Bandung yang semula berada di Kotamadya Bandung ke Wilayah hukum Kabupaten Bandung, yaitu daerah Baleendah. Peletakan batu pertamanya pada tanggal 20 April 1974, yaitu pada saat Hari Jadi Kabupaten Bandung yang ke-333. Rencana pemindahan ibu kota tersebut berlanjut hingga jabatan bupati dipegang oleh Kolonel R. Sani Lupias Abdurachman (1980-1985).

Atas pertimbangan secara fisik geografis, daerah Baleendah tidak memungkinkan untuk dijadikan sebagai ibu kota kabupaten, maka ketika jabatan bupati dipegang oleh Kolonel H.D. Cherman Affendi (1985-1990), ibu kota

Kabupaten Bandung pindah ke lokasi baru yaitu Kecamatan Soreang. Di tepi Jalan Raya Soreang, tepatnya di desa pamekaran inilah dibangun pusat pemerintahan Kabupaten Bandung seluas 24 hektare, dengan menampilkan arsitektur khas gaya priangan. Pembangunan perkantoran yang belum rampung seluruhnya dilanjutkan oleh bupati berikutnya yaitu Kolonel H.U. Djatipermana, sehingga pembangunan tersebut memerlukan waktu sejak tahun 1990 hingga 1992. Tanggal 5 Desember 2000, Kolonel H. Obar Sobarna, S.I.P. terpilih oleh DPRD Kabupaten Bandung menjadi Bupati Bandung dengan didampingi oleh Drs. H. Eliyadi Agraraharja sebagai Wakil Bupati. Sejak itu, Soreang betul-betul difungsikan menjadi pusat pemerintahan. Pada tahun 2003 semua aparat daerah, kecuali Dinas Pekerjaan Umum, Dinas Perhubungan, Dinas Kebersihan, Kantor BLKD, dan Kantor Diklat, sudah resmi berkantor di kompleks perkantoran Kabupaten Bandung hingga saat ini [2].

2.1.1 Visi

Terwujudnya Kabupaten Bandung yang Maju, Mandiri dan Berdaya Saing, melalui Tata Kelola Pemerintahan yang Baik dan Pemantapan Pembangunan Perdesaan, Berlandaskan Religius, Kultural dan Berwawasan Lingkungan

2.1.2 Misi

Untuk mewujudkan Visi di atas, maka harus ditetapkan juga Misi yang harus mendapatkan perhatian seksama dimana tugas yang diemban oleh Pemerintah Kabupaten Bandung adalah:

1. Meningkatkan profesionalisme birokrasi;
2. Meningkatkan kualitas SDM (pendidikan dan kesehatan) yang berlandaskan Iman dan takwa serta melestarikan budaya sunda;
3. Memantapkan pembangunan perdesaan;
4. Meningkatkan keamanan dan ketertiban wilayah;
5. Meningkatkan ketersediaan infrastruktur dan keterpaduan tata ruang wilayah;
6. Meningkatkan ekonomi kerakyatan yang berdaya saing;

7. Memulihkan keseimbangan lingkungan dan menerapkan pembangunan berkelanjutan.

2.1.3 Lambang dan Arti

Setiap daerah atau propinsi mempunyai logo atau lambangnya tersendiri dan setiap logo mempunyai makna atau arti dari setiap gambar atau tulisan dan di bawah ini. Berikut adalah lambang pemerintahan Kabupaten Bandung.



Gambar 2.1 Lambang Pemerintahan Kabupaten Bandung

Lambang berbentuk perisai terbagi atas empat bagian, yaitu :

1. Bagian kanan atas berlatar kuning emas

Dengan gambar gunung (Gb. Tangkuban Perahu) berwarna hijau, melambangkan bahwa Kabupaten Bandung termasyhur karena tanahnya yang subur di daerah bergunung-gunung, dan sebagai ciri memiliki gunung Tangkuban Perahu yang sangat terkenal dengan legenda Sangkuriang.

2. Bagian melintang bergerigi

Merupakan bentuk bendungan kokoh kuat berwrna hitam. Melambangkan masyarakat Kabupaten Bandung memiliki pendirian yang kokoh dan kuat, baik secara fisik dalam membendung hawa nafsu.

3. Pohon kina berwarna hijau dan berlatar belakang merah

Melambangkan di Kabupaten Bandung kaya akan air, baik air maupun air danau. Kabupaten Bandung di lintasi oleh sungai Citarum, sungai Cikapundung, dab sungai-sungai kecil lainnya. Kabupaten Bandung danau/situ Patengang, Situ Cileunca, Situ Lembang, Situ Ciburuy, dan danau-danau lainnya.

4. Dibawah perisai tertulis dalam pita kuning: REPEH RAPIH KERTARAHARJA Artinya :

REPEH : Suasana kehidupan yang aman dan tentram.

RAPI : Suasana kehidupan yang rukun dan tertib dalam lingkungan yang bersih, sehat dan asri.

KERTARAHARJA : Tatanan kehidupan yang sejahtera lahir dan batin secara seimbang, serasi adil dan merata.

2.2 Landasan Teori

Salah satu unsur terpenting dalam penelitian yang memiliki peran sangat besar dalam penelitian adalah teori. Suatu landasan teori dari suatu penelitian tertentu atau karya ilmiah sering juga disebut sebagai studi literatur atau tinjauan pustaka. Salah satu contoh karya tulis yang penting adalah tulisan itu berdasarkan riset. Melalui penelitian atau kajian teori diperoleh kesimpulan-kesimpulan atau pendapat-pendapat para ahli, kemudian dirumuskan pada pendapat baru, menurut para ahli, teori adalah:

1. Setelah masalah penelitian dirumuskan, maka langkah kedua dalam proses penelitian (kuantitatif) adalah mencari teori-teori, konsep-konsep, generalisasi-generelisasi hasil penelitian yang dapat dijadikan sebagai landasan teoritis untuk pelaksanaan penelitian.
2. Teori adalah seperangkat konstruk (konsep), definisi dan proposisi yang berfungsi untuk melihat fenomena secara sistematik, melalui spesifikasi hubungan antara variabel, sehingga dapat berguna untuk menjelaskan dan meramalkan fenomena.
3. Teori adalah generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematik.

4. Teori adalah alur logika atau penalaran, yang merupakan seperangkat konsep, definisi, dan proporsisi yang disusun secara sistematis. Secara umum, teori mempunyai tiga fungsi, yaitu untuk menjelaskan (*explanation*), meramalkan (*prediction*), dan pengendalian (*control*) suatu gejala.

2.2.1 Multimedia

Secara etimologis multimedia berasal dari kata multi (Bahasa Latin, *nouns*) yang berarti banyak, bermacam-macam, dan medium (Bahasa Latin) yang berarti sesuatu yang dipakai untuk menyampaikan atau membawa sesuatu. Kata medium dalam *American Heritage Electronic Dictionary* (1991) juga diartikan sebagai alat untuk mendistribusikan dan mempresentasikan informasi (Rachmat dan Alphone, 2005/2006).

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi (Hofstetter 2001). Multimedia sering digunakan dalam dunia hiburan. Selain dari dunia hiburan, Multimedia juga diadopsi oleh dunia Game.

Multimedia juga dapat diartikan sebagai penggunaan beberapa media yang berbeda dalam menyampaikan informasi berbentuk teks, audio, grafik, animasi, dan video. Berikut adalah definisi Multimedia menurut beberapa ahli:

1. Rosch, 1996 : Multimedia adalah Kombinasi dari komputer dan video.
2. McComick, 1996: Multimedia adalah Kombinasi dari tiga elemen: suara, gambar, dan teks.
3. Steinmetz (1995, p2): Multimedia adalah gabungan dari seminimalnya sebuah media diskrit dan sebuah media kontinu. Media diskrit adalah sebuah media dimana validitas datanya tidak tergantung dari kondisi waktu, termasuk didalamnya teks dan grafik. Sedangkan yang dimaksud dengan media kontinu adalah sebuah media dimana validitas datanya tergantung dari kondisi waktu, termasuk di dalamnya suara dan video.

4. Vaughan (2004, p1): Multimedia adalah beberapa kombinasi dari teks, gambar, suara, animasi dan video dikirim ke anda melalui komputer atau alat elektronik lainnya atau dengan manipulasi digital.

2.2.1.1 Jenis-Jenis Multimedia

Multimedia dibagi menjadi beberapa jenis berdasarkan teknik pengoprasianya. Hal tersebut dijelaskan dalam sebuah buku yang berjudul Multimedia Digital (Iwan Binanto.2010:3) membagi jenis multimedia menjadi 3 yaitu:

1. Multimedia Interaktif
   Pengguna dapat mengontrol apa dan kapan elemen-elemen multimedia akan dikirimkan atau ditampilkan.
2. Multimedia Hiperaktif
   Multimedia jenis ini mempunyai suatu struktur dari elemen-elemen terkait dengan pengguna yang dapat mengarahkannya. Dapat dikatakan bahwa multimedia jenis ini mempunyai banyak tautan atau link yang menghubungkan elemen-elemen multimedai yang ada.
3. Multimedia linear
   Pengguna hanya menjadi penonton dan menikmati prouk multimedia yang disajikan dari awal hingga akhir.

2.2.1.2 Penggunaan Mulltimedia

Multimedia dapat masuk dan menjadi alat bantu yang menyenangkan. Hal ini terjadi karena kekayaan elemen-elemen dan kemudahannya digunakan dalam banyak konten yang bervariasi. Beberapa bidang yang menggunakan multimedia adalah sebagi berikut:

1. Bisnis

Aplikasi multimedia untuk bisnis meliputi presentasi, pemasaran, periklanan, demo produk, katalog, komunikasi, di jaringan, dan pelatihan. Penggunaan multimedia akan membuat kelancaran dan kemudahan transaksi bisnis.

2. Sekolah

   Multimedia sebenarnya lebih dibutuhkan di sekolah karena multimedia dapat membuat media pembelajaran lebih lengkap dan menarik. Multimedia dapat menjadi alat pengajaran elektronik yang dapat membantu pengajar.

3. Rumah

   Multimedia dapat dimanfaatkan sebagai teman atau hiburan dirumah, misalnya game.

4. Tempat Umum

   Saat ini sudah banyak tempat-tempat umum yang memasang "kiosk", yaitu produk multimedia yang berfungsi sebagai pemberi informasi mengenai tempat yang sedang dikunjungi, kuliner, dan sebagainya.

2.2.1.3 Objek Multimedia

Banyak objek yang bisa dilibatkan dalam sebuah produk multimedia. Objek-objek tersebut disusun dengan baik agar mampu menyampaikan materi pelajaran. Beberapa objek yang dapat digunakan dalam produk multimedia pembelajaran antara lain text, grafik atau image, animasi, audio, video, dan link interaktif.

1. *Text*

   Text sudah pasti banyak dikenal semua orang. Teks menjadi dasar pengolahan kata dan kalimat serta informasi berbasis multimedia. Teks mampu menjelaskan dan menyajikan materi berupa tulisan. Penggunaan teks dalam produk multimedia perlu memperhatikan beberapa hal seperti *hypertext, Text style, Text Searching*, hingga *export* dan *import text.*

2. *Image* dan Grafik

Image dan grafik yang dimaksudkan adalah *still image* yang bisa berupa foto atau gambar. Image atau grafik ini memiliki orientasi visual dimana image mampu menyajikan fenomena yang tidak mampu dihadirkan secara langsung, contoh saja ingin memperlihatkan foto planet-planet yang tentu saja tidak mungkin diperlihatkan secara langsung. Image dan grafik bahkan mampu menyajikan data-data sehingga lebih mudah dibaca serta mampu juga menyajikan sesuatu yang abstrak yang hanya dibayangkan menjadi sebuah sajian visual.

3. Animasi

   Animasi merupakan olahan gambar yang lebih lanjut dimana gambar tersebut dapat bergerak. Animasi lebih menekankan pada pembuatan ilustrasi visual oleh manusia yang dibuat menggunakan software dengan bantuan komputer. Ilustrasi dasar dibuat kemudian ilustrasi dasar tersebut dibuat bergerak sesuai dengan narasi yang telah direncanakan. Konsep animasi ini banyak digunakan untuk pembuatan film kartun serta simulasi visual sebuah fenomena yang memiliki proses atau tahapan.

4. Audio

   Audio mampu menyajikan materi pembelajaran dalam bentuk suara atau lisan. Materi pelajaran dapat direkam lalu disusun sesuai narasi kemudian dapat diperdengarkan pada peserta didik. Penggunaan audio terutama musik pada multimedia dapat menekankan sebuah gambar atau video kejadian tertentu, juga dapat digunakan sebagai background sound effect.

5. *Video*

   Video juga merupakan gambar yang bergerak, namun video kebanyakan berupa penggabungan hasil tangkapan kamera serta juga dapat menggunakan animasi yang digabungkan mengikuti narasi atau *storyboard.* Dapat juga berupa slide-slide yang disusun sesuai urutan kemudian

diconvert atau *diexport* menjadi video sehingga mampu menampilkan materi pelajaran yang dikehendaki.

6. *Link* Interaktif

   Interactive Link ini mampu menjadikan produk multimedia yang dihasilkan menjadi media pembelajaran interaktif. *Link* interaktif ini mampu menghubungkan satu objek atau tampilan layar dengan objek atau tampilan layar lainnya. *Interactive Link* juga mampu digunakan untuk menjalankan perintah tertentu sehingga produk multimedia dapat diprogram menjadi media pembelajaran interaktif dimana peserta didik mampu secara aktif berinteraksi dengan media pembelajaran berbasis multimedia tersebut. Menurut Luther dalam Sutopo (2012: 111) penggunaan link interaktif dalam produk multimedia dapat berupa button (tombol), menu, dan *hypertext*.

2.2.1.4 Objek 2D

Objek grafik 2 dimensi adalah sekumpulan titik-titik 2 dimensi yang dihubungkan dengan garis lurus, baik berupa *polyline, polygon* atau *kurva*. Objek grafik 2 dimensi didefinisikan sebagai kumpulan titik 2 dimensi yang secara komputasi dinyatakan sebagai array 1D, atau linkedlist sesuai dengan struktur data yang digunakan dalam menyatakan kumpulan titik 2D ini. Sedangkan Grafik komputer 2D adalah sebuah generasi gambar digital berbasis komputer, yang kebanyakan mengambil objek-objek dua dimensi (2D). Model Grafik 2D merupakan kombinasi dari model geometri (juga disebut sebagai grafik vektor), gambar digital (*raster graphics*), fungsi matematika, dan sebagainya. Komponen-komponen ini dapat dimodifikasi dan dimanipulasi oleh transformasi geometri dua dimensi, seperti translasi, rotasi, dan dilatasi.

Model-model yang digunakan pada disain grafis 2D biasanya tidak mendukung bentuk-bentuk tiga-dimensi, atau fenomena yang bersifat tiga dimensi, seperti pencahayaan, bayangan, pantulan, refraksi, dan sebagainya. Namun demikian, mereka dapat membuat model berlapis-lapis (*layer*); nyata, translusen, dan transparan, yang dapat ditumpuk dalam urutan tertentu. Urutan tersebut

biasanya didefinisikan dengan angka (kedalaman lapisan, atau jarak dari si penglihat).

2.2.1.5 Objek 3D

3 dimensi biasa disebut 3D adalah bentuk dari benda yang memiliki panjang, lebar, dan tinggi. Grafik 3 Dimensi merupakan teknik penggambaran yg berpatokan pada titik koordinat sumbu x(datar), sumbu y(tegak), dan sumbu z(miring).Representasi dari data geometrik 3 dimensi sebagai hasil dari pemrosesan dan pemberian efek cahaya terhadap grafika komputer 2D. Tiga Dimensi, biasanya digunakan dalam penanganan grafis. 3D secara umum merujuk pada kemampuan dari sebuah *video card* (*link*). Grafik 3D merupakan perkembangan dari grafik 2D. Didalam grafika komputer, 3D merupakan bentuk grafik yang menggunakan representasi data geometri tiga dimensi.

Suatu objek rangka 3D apabila disinari dari arah tertentu akan membentuk bayangan pada permukaan gambar. Proses pembuatan grafik komputer 3D dapat dibagi ke dalam tiga fase, yaitu 3D modeling yang mendeskripsikan bentuk dari sebuah objek, layout dan *animation* yang mendeskripsikan gerakan dan tata letak sebuah objek, dan 3D rendering yang memproduksi image dari objek tersebut. Istilah atau Pengertian Grafik 3D adalah sebuah gambar, garis, lengkungan dan sebagainya yang memiliki titik-titik yang menghubungkan menjadi sebuah bentuk 3D. Di dalam dunia game, 3D secara umum merujuk pada kemampuan dari sebuah video card *(link).* Saat ini video *card* menggunakan variasi dari instruksi-instruksi yang ditanamkan dalam video *card* itu sendiri (bukan berasal dari *software*) untuk mencapai hasil grafik yang lebih realistis dalam memainkan game komputer.

2.2.2 Modelling

Modeling merupakan istilah umum untuk menunjukkan terjadinya proses belajar melalui pengamatan dari orang lain dan perubahan yang terjadi karenanya melalui peniruan. Teknik modeling bukan sekedar menirukan atau mengulangi apa yang dilakukan orang model (orang lain), tetapi modeling melibatkan penambahan

dan atau pengurangan tingkah laku yang teramati, menggenalisir berbagai pengamatan sekalligus, melibatkan proses *kognitif* (Alwisol, 2009:292).

Jenis-Jenis Teknik *Modeling* Menurut Willis (2004 : 78) ada 2 yaitu [3]:

1. *Social Modeling*

   Teknik yang membentuk perilaku baru melalui model sosial dengan cara imitasi observasi.

2. *Self Modeling*

   Yaitu teknik yang bertujuan menghilangkan perilaku tertentu, dimana konselor menjadi model, dan klien berjanji akan mengikuti.

2.2.2.1 3D Modelling

3D *modelling* adalah proses mengembangkan matematika representasi dari setiap tiga-dimensi benda (baik benda mati atau hidup) melalui perangkat lunak khusus. Produk ini disebut sebagai model 3D. Hal ini dapat ditampilkan sebagai gambar dua dimensi melalui proses yang disebut 3D rendering atau digunakan dalam komputer simulasi fenomena fisik. Model juga dapat secara fisik dibuat menggunakan perangkat Printing 3D. Model dapat dibuat secara otomatis atau manual. Manual proses pemodelan geometris mempersiapkan data untuk komputer grafis 3D mirip dengan seni plastik seperti mematung [4].

2.2.2.2 Manfaat 3D *Modeling*

3D *modeling* banyak memiliki manfaat didalam berbagi bidang sebagai contoh, yaitu:

1. Architecture Design

Jika Anda seorang Arsitek, maka 3D akan sangat membantu Anda dalam menuangkan ide-ide kreatif design dengan sangat cepat dan mudah. Berbeda bila hanya mengandalkan kemampuan konvensional seperti sketsa yang mempunyai keterbatasan dalam memandang ke *view* 3 dimensi, maka dengan 3D Anda akan dapat melihat bentuk rancangan bangunan Anda dari segala sisi pandangan. Disamping itu dengan 3D, Anda akan dapat memberikan *texture* maupun *lighting* menyerupai seperti kondisi aslinya kelak bangunan tersebut

terbangun, sehingga akan mudah bagi *Client* Anda untuk mengerti dan memahami design yang Anda buat. Jika Anda bukan Arsitek pun, tidak usah kuatir, *skill* 3D yang Anda miliki dapat menjadikan Anda sebagai seorang *Visualizer* yang dapat menjadi pendamping bagi seorang Arsitek dalam membuat visual bangunan yang akan dirancang.

2. *Interior Design*

   Sama halnya seperti Arsitek, *designer Interior* pun juga akan sangat terbantu dengan teknologi 3D. Sebagai seorang *Visualizer* pun, Anda dapat menjadi pendamping dari seorang sang *designer Interior* dalam menvisualkan rancangan ruang-ruang yang akan diwujudkan nantinya. Dengan 3D, setiap ruang yang dirancang akan dapat divisualkan mendekati ruang yang sebenarnya. Layaknya bermain *game* the Sims, Anda dapat menata *furniture*, elemen dekoratif *interior* lainnya seperti lukisan, foto, patung dan lainnya dengan sangat leluasa dan mudah.

3. *Cartoon Design*

   Kalau dulu kartun (*cartoon*) dibuat dengan sketsa tangan lalu meningkat ke aplikasi *software* 2D, sekarang kartun menapaki era baru dengan teknologi 3D. Trend ini semakin meningkat, dikarenakan dengan 3D, *cartoon* akan semakin tampak hidup dan nyata, beratraksi menghibur penonton yang melihatnya. Aplikasi 3D saat ini, sudah semakin memudahkan para designer dan animator, bahkan yang pemula sekalipun untuk membuat kartun 3D, karena sudah banyak fitur-fitur penunjang pembuat kartun yang disediakan oleh setiap aplikasi 3D.

4. *Logo Design*

   Tidak hanya *cartoon*, *design* dan animasi logo pun saat ini sudah memasuki era 3D. Artinya, logo-logo saat ini sudah banyak yang ditampilkan dengan bentuk dan tampilan *real* 3D. Sehingga logo terlihat lebih hidup, menarik dan tampil lebih memikat. Aplikasinya dapat dijumpai dibanyak hal, seperti: dibidang *broadcast/tv*, *advertising*, perkantoran, industri, *branding*/merk, percetakan, dan sebagainya. Intinya setiap perusahaan (*company*) dan juga

produk-produk sangat membutuhkan logo sebagai citra visual mereka dan 3D merupakan solusi tepat dalam menvisualkannya.

5. *Product Design*

   Product design, mencakup berbagai barang seperti : *product* industri, pecah belah, mainan, computer/it, alat komunikasi dan sebagainya, telah menjadikan 3D sebagai standar dalam menvisualkan design terbaru dari setiap produk yang akan dirilis dipasaran. Ambil contoh misalkan : sebuah perusahaan pembuat laptop yang akan mengeluarkan sebuah design laptop terbaru, maka bentuk design, material, tampilannya akan terlebih dahulu diwujudkan dengan media 3D. Sehingga bila telah ok dan disetujui, tinggal dibuat kedalam manufakturnya. Ini tentu saja akan menghemat cost anggaran disamping juga dapat menghemat waktu dan tenaga produksi.

6. *Game Design*

   Saat ini hampir semua *game* sudah ditampilkan dalam bentuk 3D. Ini dimaksudkan agar game dapat tampil lebih atraktif, hidup dan memikat hati para *gamer*. Dengan 3D, para gamer akan mendapatkan sensasi bermain *game* yang luar biasa, yang berdampak pada semakin meningkatnya minat, animo maupun antusiasme dalam memainkan *game*.

7. *Furniture Design*

   Dibidang *furniture design* sudah banyak yang menggunakan teknologi 3D dalam merancang dan menampilkan bentuk dari suatu *design furniture* terbaru. Pembuatan *design* yang detail dan rumit sekalipun dapat diselesaikan dengan pendekatan aplikasi 3D. Begitu juga dengan penerapan material yang dapat mendekati material aslinya sehingga *design* akan tampil lebih *real* dan hidup

2.2.2.3 Texturing

Proses ini adalah proses pembuatan dan pemberian warna dan material (*texture*) pada objek yang telah dimodelkan sebelumnya sehingga akan tampak suatu kesan yang nyata. Pemberian material atau *texture* pada objek 3D akan mendefinisikan rupa dan jenis bahan dari objek 3D. Material atau *texture* dapat berupa foto atau gambar yang dibuat dengan *aplikasi software* 3D, seperti 3ds max,

Maya, dan lain-lain, atau dengan bantuan *software digital imaging*, seperti *photoshop, photoPaint*, atau *Gimp*.

2.2.2.4 lighting

Lighting adalah proses pembuatan dan pemberian cahaya pada model sehingga diperoleh kesan visual yang *realistis* karena terdapat kesan kedalaman ruang dan pembayangan (*Shadow*) objek. Tanpa adanya *lighting*, maka objek 3D anda menjadi tidak menarik dan juga tidak *realistis*.

2.2.2.5 Animation

Animation adalah proses pembuatan animasi untuk model. Animasi dapat berupa gerakan, baik itu gerakan objek/model atau gerakan kamera untuk menciptakan animasi *walktought*, animasi *flythrough*, dan lain-lain. Anda dapat menentukan arah dimulainya suatu gerakan animasi yang, tentu saja disesuaikan dengan *storyboard* yang telah dibuat pada tahap *pre production*

2.2.2.6 Rendering

Rendering adalah proses menangkap segala sesuatu yang ada di dalam *scene* termasuk geometri, material, cahaya, lingkungan serta efek-efek yang terdapat didalamnya. Dalam sebuah gambar *output* dari *scene* 3D atau suatu object, Fitur-fitur seperti *materials*, *lighting*, *oversampling* dan *shadows* memiliki pengaruh dalam efek dan kualitas hasil *rendering*. Semakin banyak fitur yang ditambahkan , maka semakin realistik hasilnya, akan tetapi akan mempengaruhi lama waktu *rendering.Rendering* telah digunakan dalam arsitektur,*video game*, simulator, film dll. Dalam hal grafis 3D, *rendering* bisa dilakukan perlahan-lahan, seperti pada pra-rendering, atau secara *real-time*. *Pra-rendering* adalah proses komptasi secara intensif yang biasanya digunakan untuk pembuatan film, sedangkan *real-time rendeering* sering dilakukan untuk 3D *video game* yang mengandalkan pada penggunaan kartu grafis dengan *akselerator* 3D *hardware* [5].

2.2.2.6.1 Metode Rendering

Metode *rendering* yang paling sederhana dalam grafika 3D [5]:

1. *Wireframe rendering*

   Wireframe yaitu Objek 3D dideskripsikan sebagai objek tanpa permukaan. Pada *wireframe rendering*, sebuah objek dibentuk hanya terlihat garis-garis yang menggambarkan sisi-sisi *edges* dari sebuah objek. Metode ini dapat dilakukan oleh sebuah komputer dengan sangat cepat, hanya kelemahannya adalah tidak adanya permukaan, sehingga sebuah objek terlihat *tranparent*. Sehingga sering terjadi kesalahpahaman antara siss depan dan sisi belakang dari sebuah objek.

2. *Hidden Line Rendering*

   Metode ini menggunakan fakta bahwa dalam sebuah objek, terdapat permukaan yang tidak terlihat atau permukaan yang tertutup oleh permukaan lainnya. Dengan metode ini, sebuah objek masih direpresentasikan dengan garis-garis yang mewakili sisi dari objek, tapi beberapa garis tidak terlihat karena adanya permukaan yang menghalanginya. Metode ini lebih lambat dari dari *wireframe rendering*, tapi masih dikatakan relatif cepat. Kelemahan metode ini adalah tidak terlihatnya karakteristik permukaan dari objek tersebut, seperti warna, kilauan (*shininess*), tekstur, pencahayaan, dll.

3. *Shaded Rendering*

   Pada metode ini, komputer diharuskan untuk melakukan berbagai perhitungan baik pencahayaan, karakteristik permukaan, *shadow casting*, dll. Metode ini menghasilkan citra yang sangat realistik, tetapi kelemahannya adalah lama waktu *rendering* yang dibutuhkan

2.2.3 Peta

Menurut Erwin Raisz,[6] pengertian peta adalah gambaran konvensional permukaan bumi yang diperkecil dengan berbagai kenampakan dan ditambah tulisan-tulisan sebagai tanda pengenal. Pengertian peta secara umum, mengartikan peta sebagai gambaran sebagian atau seluruh wilayah permukaan bumi dengan berbagai kenampakan pada suatu bidang datar yang diperkecil dengan menggunakan skala tertentu.

Berbeda dengan pengertian umum, Organisasi Katografi Internasional (*International Cartographic Association* [ICA]) memberikan definisi, pengertian peta merupakan sebagai suatu gambaran atau representasi unsur-unsur kenampakan abstrak permukaan Bumi dan benda-benda angkasa. Satu hal yang berbeda, definisi dari ICA, melepaskan definisi peta yang terikat pada gambaran muka Bumi, dan meluaskannya pada penggambaran objek-objek di ruang angkasa. Hal ini dibutuhkan karena perkembangan ilmu pengetahuan, teknologi, dan sains, telah membentangkan sayapnya hingga ke tempat-tempat lain di luar Bumi.

2.2.3.1 Sejarah Peta

Sejarah pembuatan peta di dunia bermula ketika para petualang masa lalu bila menjumpai orang di suatu tempat dan bertanya tentang arah jalan, biasanya orang tersebut segera menggores tanah dengan menggunakan sepotong kayu. Itulah awal dari sejarah pembuatan peta pertama di dunia. Akan tetapi, peta paling awal yang menggambarkan penampakan pada bidang datar dibuat oleh bangsa Babilonia sekitar 2.300 SM. Peta tertua tersebut berupa papan tulis batu berukuran kecil dari tanah liat. Peta tua lainnya dibuat oleh penduduk Pulau Marshall di kawasan Oseania. Peta ini berupa anyaman serabut rotan yang diatur sedemikian rupa untuk menunjukkan penempatan pulau[6].

Sejarah pembuatan peta terus berlanjut ditunjang oleh ilmu hitung (matematika) dan ilmu-ilmu lain tumbuh dan berkembang. Rasa ingin tahu dan jarak capai yang ditempuh manusia semakin besar. Pengamatan dan pengukuran bumi secara sederhana mulai dilakukan sehingga muncul peta pertama yang

menghadirkan dunia. Ilmuwan Yunani yang cukup berjasa memetakan dunia di antaranya Anaximander dan Eratosthenes. Sekitar tahun 150 SM, telah terbit peta dunia berbentuk kerucut yang telah menggunakan pengukuran yang agak cermat. Peta tersebut dibuat seorang ahli geografi ternama yang bernama Ptolemeus. Ia dianggap sebagai Bapak Kartografi.

Pengetahuan pembuatan peta terus berkembang. Abad ke-15 sampai 17 merupakan era perpetaan. Para kartografer Belanda, Portugis, Spanyol, Italia, dan Jerman berjibaku memetakan wilayah-wilayah yang akan diarungi para petualang. Saat itu memang bangsa-bangsa di Eropa tengah berlomba mencari wilayah-wilayah baru untuk dikuasainya, terutama daerah penghasil rempah-rempah, seperti kepulauan Nusantara. Di Abad itu, banyak peta kuno dibuat kendati minumnya peralatan. Daya imajinasi kartografer memegang peranan penting sehingga penentuan arah utara dan selatan masih kacau balau, mata angin kadang terbalik, dan skala peta tidak proporsional (sebanding/seimbang). Meskipun demikian, peta-peta kuno saat itu memiliki mutu artistik (nilai seni) tingga serta kualitas percetakan dan pewarnaan yang cukup baik. Salah satu contoh peta kuno itu adalah peta Asia Tenggara buah karya Willem Blaeu, seorang kartografer Belanda. Pada abad ke-18 sampai dengan 19, Negara-negara di Eropa dan Amerika Serikat mulai beramai-ramai memetakan negerinya. Di Eropa, Prancis memelopori survey topografi nasional sejak tahun 1793. Inggris, Spanyol, Austria, Swiss, dan Negara-negara lain segera mengikuti langkah Prancis. Begitu juga Amerika Serikat melakukan pemetaan secara besar-besaran di seluruh Negara bagiannya sejak tahun 1879. Negeri ini bahkan berhasil menyelenggarakan Kongres Geografi Internasional pada tahun 1891 yang menyepakati pemetaan ke seluruh dunia dengan skala 1: 1000.000.

Sepanjang abad ke-20 telah muncul upaya-upaya pembaharuan teknis dalam pemetaan. Pemotretan dari udara mulai dikembangkan secara ekstensif (menjangkau secara luas) selama Perang Dunia I dan II. Kemudian, pada tahun 1966, Amerika Serikat mampu meluncurkan satelit Pageos dan Satelit-satelit lain pada tahun-tahun berikutnya. Hal ini diikuti oleh Uni Soviet. Melalui satelit-satelit

tersebut berhasil dikirim foto video beberapa bagian muka bumi ke stasiun di bumi. Foto-foto itu lalu diubah menjadi peta yang lebih rinci dan akurat.

2.2.3.2 Manfaat Peta

Peta dapat dimanfaatkan untuk berbagai kepentingan, yaitu:

1. Mengetahui jarak satu tempat dengan tempat lainnya. Dengan jarak antara tempat yang menggunakan skala peta, kalian dapat menghitung satu dengan tempat lainnya di muka bumi.
2. Mengetahui arah suatu tempat. Contoh: dengan peta Kawasan ASEAN , kita dapat mengetahui bahwa Negara Indonesia berada disebelah selatan Negara Filipina. Negara Brunei Darussalam berada di sebelah utara Kalimantan. Pulau Sulawesi berada di sebelah timur Pulau kalimantan.
3. Peta dapat digunakan untuk menjelaskan kondisi lingkungan suatu tempat. Contoh: melalui peta dapat diketahui suatu wilayah berada di daerah tropis, daerah kutub, atau daerah sedang. Dengan mengetahui bahwa Antartika berada di Kutub Selatan kita dapat mengungkapkan bahwa di tempat itu suhu udaranya sangat dingin dan dimana-mana terdapat tumpukan salju. Melalui warna pada peta kita juga dapat mengetahui suatu wilayah berupa daerah datar atau bergunung-gunung. Contoh: pada peta rupa bumi, daerah dataran rendah digambar dengan warna hijau dan daerah pegunungan digambar dengan warna coklat.
4. Melalui peta tematik kita dapat memperoleh data. Contoh: dari peta kepadatan penduduk kita dapat memperoleh data provinsi-provinsi mana saja yang penduduknya masih jarang dan provinsi mana yang penduduknya sangat padat.
5. Melalui peta orang dapat memperkirakan kemungkinan usaha yang dilakukan. Bila kalian akan membuka usaha pertambakan maka lokasi (tempat) usaha yang dipilih adalah di tepi laut. Bila kita ingin membuka usaha kebun bunga, maka tempat usaha yang dipilih adalah daerah pegunungan. Tempat semacam itu hanya dapat diketahui melalui peta.

2.2.3.3 Jenis Peta

Berikut ini adalah penjelasan peta berdasarkan Jenisnya, jenis peta dibagi mejadi 3 yaitu :

1. Jenis peta berdasarkan isinya

   Jenis Peta berdasarkan isinya yaitu sebagai berikut :

   a. Peta Umum

      Peta umum adalah peta yang menggambarkan seluruh bentuk kenampakan alam yang ada di permukaan bumi, baik kenampakan asli maupun kenampaka buatan.

      Peta Umum di bedakan lagi menjadi tiga jenis, yaitu:

      1. Peta Dunia

         Peta dunia adalah peta yang mengfgambarkan bentuk dan letak muka bumi serta wilayah setiap negara di dunia dengan skala tertentu.

      2. Peta Topografi

         Atau biasa disebut peta rupa bumi yaitu peta yang menggambarkan permukaan bumi beserta tinggi rendahnya.

      3. Peta Korografi

         Peta korografi adalah peta yang menggambarkan permukaan bumi, baik sebagian maupun seluruhnya yang bercorak luas dan dengan skala kecil.

      4. Peta Khusus

         Peta khusus adalah peta yang menggambarkan suatu kenampakan alam tertentu yang ada dipermukaan bumi.

2. Jenis peta berdasarkan bentuknya

   Jenis peta berdasar bentuknya dapat dibedakan menjadi:

   a. Peta Digital

      Peta yang digambarkan pada sebuah aplikasi komputer, biasanya menggunakan Sistem Informasi Geografis (SIG).

   b. Peta Timbul (*relief*)

Peta timbul atau relief adalah peta yang menggambarkan bentuk sebenarnya dari permukaan bumi.

c. Peta Datar

Peta datar adalah peta yang digambarkan dalam bidang datar berbentuk dua dimensi.

3. Jenis peta berdasarkan skalanya

Berdasarkan skalanya, peta dapat di bagi menjadi:

a. Peta Kadaster / Teknik

Peta kadaster atau teknik adalah peta yang memiliki skala antara 1:100 hingga 1:5.000. Banyak dipakai oleh Departemen Dalam Negeri dan Dinas Agraria (Badan Pertanahan Nasional).

b. Peta Skala Besar

Peta ini memiliki skala antara 1:5.000 hingga 1:250.000 yang digunakan untuk menggambarkan daerah dengan skala sempit, seperti peta kecamatan.

c. Peta Skala Menengah

Memiliki skala antara 1:250.000 hingga 1:500.000 yang digunakan untuk menggambarkan daerah yang agak lyuas , seperti peta provinsi.

d. Peta Skala Kecil

Memiliki skala antara 1:500.000 hingga 1:1.000.000 atau lebih yang digunakan untuk menggambarkan daerah yang relatif luas , seperti benua atau dunia.

2.2.3.4 Rute

Rute adalah lintasan atau jalan yang akan dilalui seseorang atau kelompok dengan arah-arah tertentu untuk menuju suatu tempat yang telah direncanakan.

2.2.3.5 Lintasan Terpendek (*Shortest Path*)

Shortest path adalah pencarian rute atau path terpendek antara *node* yang ada pada *graph*.biaya (*cost*) yang dihasilkan adalah minimum.

2.2.4 Algoritma

Algoritma berasal dari *al-kwarizmi* yang terdapat dibuku Abu Ja'far Muhammad Ibnu Musa Al-Kwarizmi seorang ahli matematika dari persia. Dalam buku ini definisi algoritma adalah susunan langkah penyelesaian suatu masalah secara sistematika dan logis [4].

2.2.4.1 Algoritma Pencarian Rute Terdekat

Lintasan terpendek merupakan salah satu masalah yang dapat diselesaikan dengan menggunakan *graph*. Jika diberikan sebuah graph berbobot, masalah lintasan terpendek adalah bagaimana kita mencari sebuah jalur pada *graph* yang meminimumkan jumlah bobot sisi pembentuk jalur tersebut. Algoritma-Algoritma dalam *Shortest Path* ada beberapa macam yaitu :

2.2.4.2 Pencarian Menggunakan Algortima *Heuristik*

Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namum dengan kemungkinan mengorbankan kelengkapan (completeness). Fungsi heuristik digunakan untuk mengevaluasi keadaankeadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

Contoh algoritma pencarian heuristik yaitu:

1. Algoritma *Greedy*
2. Algoritma *Dijkstra*
3. Algoritma *Hill Climbing*

2.2.4.3 Algoritma *Greedy*

Algoritma *Greedy* adalah algoritma yang memecahkan masalah langkah demi langkah, pada setiap langkah dilakukan dengan cara [5]:

1. Mengambil pilihan yang terbaik yang dapat diperoleh saat itu
2. Berharap bahwa dengan memilih optimum local pada setiap langkah akan mencapai optimum global

Berikut adalah tahapan langkah-langkah algoritma *greedy* untuk mencari rute terpendek :

1. Menentukan titik awal dan titik tujuan, misalnya titik awal a.
2. Perikasa semua sisi yang langsung bersisian dengan titik a. Pilihsisi yang bobotnya terkecil. Sisi ini menjadi lintasan terpendek pertama, sebut saja L(1).
3. Tentukan lintasan terpendek kedua dengan cara berikut:
4. Hitung: d(i) = panjang L(1) + bobot sisi dari simpul akhir L(1) kesimpul i yang lain.
5. Pilih d(i) yang terkecil.
6. Bandingkan d(i) dengan bobot sisi (a, i). Jika bobot sisi (a, i) lebihkecil daripada d(i), maka L(2) = L(1) U (sisi dari simpul akhir L(i)ke simpul i)
7. Dengan cara yang sama, ulangi langkah 2 untuk menentukanlintasan terpendek berikutnya.

Pseudocode Algoritma *Greedy*

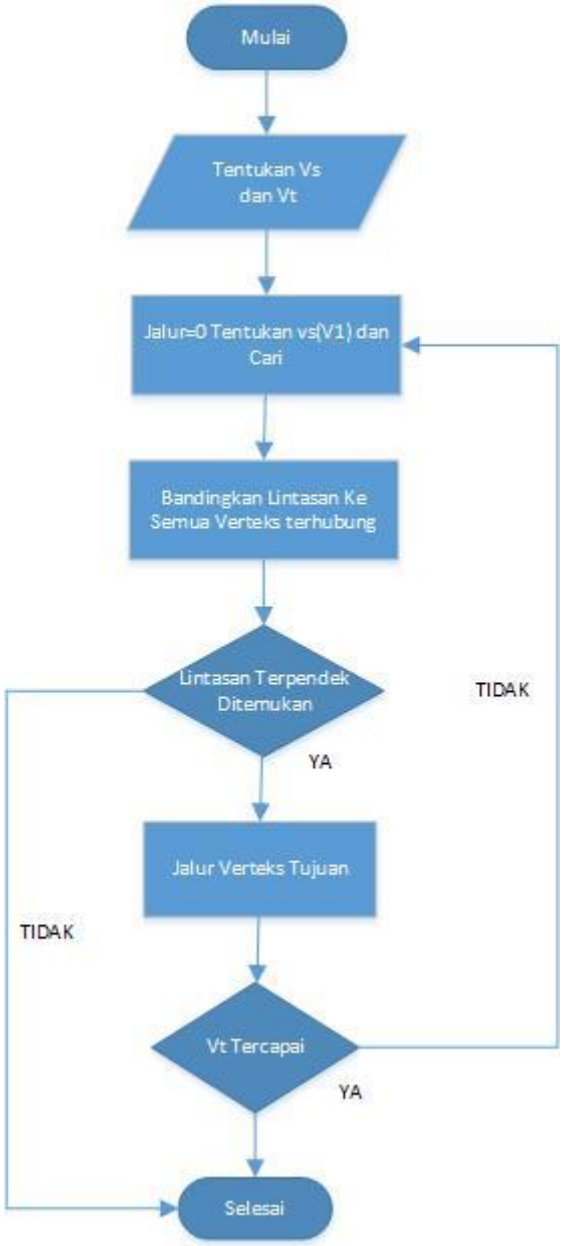
Berikut adalah *pseudocode* algoritma *greedy* yang diperlukan untuk melakukan proses pencarian jalur terpendek:

```
Procedure greedy (input c: himpunan_kandidat'
Output s: himpunan_solusi)
    { menentukan sosial optimum dari persoalan optimasi dengan algoritma greedy
        Masukan: himpunan kandidat c
        Keluaran: himpunan solusi s
    }
Deklarasi
X: kandidat;
Algoritma;
S ← {} { inisialisasi S dengan kosong }
```

```
While (belum SOLUSI (S) ) and (C= {}) do
X ← SELEKSI (C); {pilih kandidat dari C}
C ← C  - {X}
        If LAYAK (S U {X} then
                S ← S U {X}
        Endif
Endwhile
{ solusi (S) sudah diperoleh or C = {} }
```

Flowchart Algoritma *Greedy*

Beritkui adalah *flowchart* algoritma *greedy* untuk melakukan proses pencarian jalur terpendek:



Gambar 2.2 *Flowchart* Algoritma *Greedy*

Kelebihan dan Kekurangan Algoritma *Greedy*

Prinsip pencarian lintasan terpendek memakai fungsi ”Seleksi” dan itu berguna untuk menentukan jalan tersingkat untuk menuju suatu tempat. Sehingga, dapat sampai tepat waktu menuju tempat tujuan. Hasil analisis berdasarkan bobot-bobot yang berbeda, menunjukkan bahwa semakin banyak bobot yang diberikan, maka semakin akurat pula datayang dihasilkan. Sehingga menghasilkan waktu yang efisien.

Sedangkan kekurangan dari algoritma *greedy* adalah sebagai berikut:

1. Algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada (sebagaimana pada metode *exhaustive search*).
2. Pemilihan fungsi SELEKSI: Mungkin saja terdapat beberapa fungsi SELEKSI yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma bekerja dengan benar dan menghasilkan solusi yang benar-benar optimum. Karena itu, pada sebagian masalah algoritma *Greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum.

2.2.4.4 *Hill Climbing*

Metode ini hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin(Sri Kusumadewi 2003, h. 34).

Metode *hill climbing* terinspirasi akan langkah-langkah yang dilakukan oleh para pendaki dalam menemukan camp mereka yang terletak diatas lereng gunung bagian atas. Para pendaki salalu akan mencari jalan yang lebih pintas untuk mencapai tujuannya. Diilustrasikan bahwa untuk mencapai titik E dari titik A mempunyai 3 alternatif jalur yaitu A – B – D – E, A – C – D – E, dan A – D – E.

Penentuan rute yang dipilih pada metode *Hill Climbing* akan dibandingkan ketiga jalur tersebut mana yang paling sedikit cost yang arus dikeluarkan, apakah rute yang pling pendek ataupun tinngkat kemacetan yang paling kecil, pemilihan akan bergantung pada informasi yang diberikan pada peta yang akan dilalui.

Berikut adalah langkah-langkah yang diperlukan untuk *Hill Climbing Search* adalah sebagai berikut :

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kerjakan langkah-langkah berikut sampai solusinya ditemukan, atau sampai tidak ada node baru yang akan diaplikasikan pada keadaan sekarang :
    a. Cari node yang belum pernah digunakan; gunakan node ini untuk mendapatkan keadaan yang baru.
    b. Evaluasi keadaan baru tersebut.
        1. Jika keadaan baru merupakan tujuan, keluar.
        2. Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
        3. Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan pencarian.

Pseudocode Algoritma *Hill Climbing*

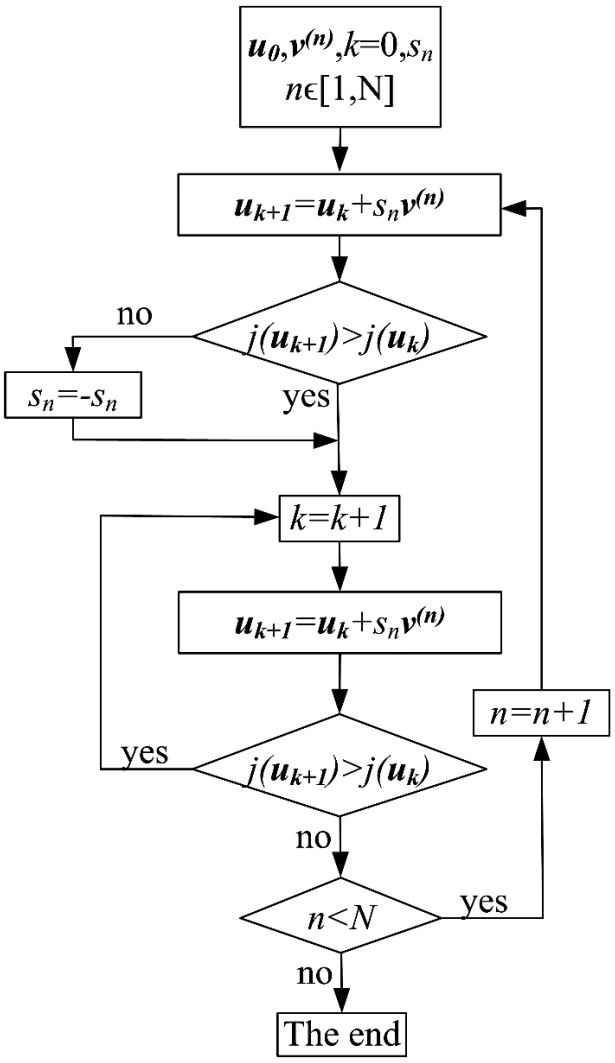
Berikut adalah *pseudocode* algoritma *hill climbing* yang diperlukan untuk melakukan proses pencarian jalur terpendek :

```
Discrete Space Hill Climbing Algorithm
   currentNode = startNode;
   loop do
      L = NEIGHBORS(currentNode);
```

```
      nextEval = -INF;
      nextNode = NULL;
      for all x in L
         if (EVAL(x) > nextEval)
              nextNode = x;
              nextEval = EVAL(x);
      if nextEval <= EVAL(currentNode)
         //Return current node since no better neighbors exist
         return currentNode;
      currentNode = nextNode;
Continuous Space Hill Climbing Algorithm
   currentPoint = initialPoint;    // the zero-magnitude
vector is common
   stepSize = initialStepSizes;    // a vector of all 1's is
common
   acceleration = someAcceleration; // a value such as 1.2 is
common
   candidate[0] = -acceleration;
   candidate[1] = -1 / acceleration;
   candidate[2] = 0;
   candidate[3] = 1 / acceleration;
   candidate[4] = acceleration;
   loop do
      before = EVAL(currentPoint);
      for each element i in currentPoint do
         best = -1;
         bestScore = -INF;
         for j from 0 to 4         // try each of 5 candidate
locations
            currentPoint[i] = currentPoint[i] + stepSize[i] *
candidate[j];
            temp = EVAL(currentPoint);
            currentPoint[i] = currentPoint[i] - stepSize[i] *
candidate[j];
            if(temp > bestScore)
                 bestScore = temp;
                 best = j;
         if candidate[best] is not 0
            currentPoint[i] = currentPoint[i] + stepSize[i] *
candidate[best];
            stepSize[i] = stepSize[i] * candidate[best]; //
accelerate
      if (EVAL(currentPoint) - before) < epsilon
         return currentPoint;
```

Flowchart Algoritma *Hill Climbing*

Berikut ini adalah *flowchart* algoritma *Hill Climbing* untuk melakukan proses pencarian jalur terpendek :



Gambar 2.3 Flowchart *Algoritma* Hill Climbing

Berikut adalah kelebihan dari algoritma *Hill Climbing* :

1. Ruang masalah dapat didefinisikan sebagai *initial state* (keadaaan awal) menuju keadaan *goal state* (tujuan) dengan ketentuan perulangan dan percabangan sesuai aturan *Hill Climbing.*

2. Aturan produksinya dapat mendefinisikan state ke state lain beserta perubahannya.
3. Menghasilkan solusi evaluasi kedaan yang lebih baik dari keadaan semula.
4. Kombinasi dari metode *depth-first* search dan metode *breadth-first* search dengan mengambil kelebihan dari kedua metode tersebut.

Kekurangan dari *Hill Climbing* adalah sebagai berikut:

1. Tidak diperbolehkan untuk kembali ke *node* pada level yang lebih rendah meskipun node pada level yang lebih rendah tersebut memiliki nilai *heuristik* yang lebih baik
2. Untuk mengimplementasikan metode *Hill Climbing* menggunakan *graph* keadaan, dibutuhkan 2 antrian yang berisi node-node seperti berikut :
   a. *Open*, berisi *node* yang dibangkitkan, namun belum diuji. Berupa antrian berprioritas yang berisi elemen-elemen dengan nilai *heuristik* tertinggi.
   b. *Closed*, berisi *node-node* yang sudah diuji. Jadi harus menunggu sampai node teruji membutuhkan waktu yang lama dan akurasi yang kurang baik.

2.2.4.5 Algoritma *Dijkstra*

Algoritma *Dijkstra*, dinamai menurut penemunya, Edsger Dijkstra, merupakan salah satu varian dari algoritma *greedy*, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan lempang (*straight-forward*). Sesuai dengan artinya yang secara harfiah berarti tamak atau rakus (namun tidak dalam konteks negatif), algoritma *greedy* ini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan. Prinsipnya, ambillah apa yang bisa Anda dapatkan saat ini (*take what you can get now!)*, dan keputusan yang telah diambil pada setiap langkah tidak akan bisa diubah kembali [6].

Ada beberapa kasus pencarian lintasan terpendek yang diselesaikan menggunakan algoritma *Dijkstra*, yaitu: pencarian lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*), pencarian lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*), pencarian lintasan terpendek dari

simpul tertentu ke semua simpul yang lain (*single-source shortest path*), serta pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*).Intinya, algoritma *greedy* ini berupaya membuat pilihan nilai optimum lokal pada setiap langkah dan berharap agar nilai optimum lokal ini mengarah kepada nilai optimum global.

Langkah-langkah dalam menentukan lintasan terpendek pada algoritma *Dijkstra* yaitu:

1. Pada awalnya pilih titik dengan bobot yang terendah dari titik yang belum terpilih, diinisialisasikan dengan “0“ dan yang sudah terpilih diinisialisasikan dengan “1“.
2. Bentuk tabel yang terdiri dari titik, status, bobot dan *predecessor*. Lengkapi kolom bobot yang diperoleh dari jarak titik sumber ke semua titik yang langsung terhubung dengan titik sumber tersebut.
3. Jika titik sumber ditemukan maka tetapkan sebagai titik terpilih.
4. Tetapkan titik terpilih dengan label permanen dan perbarui titik yang langsung terhubung.
5. Tentukan titik sementara yang terubung pada titik yang sudah terpilih sebelumnya dan merupakan bobot terkecil dilihat dari table dan tentukan sebagai titik terpilih berikutnya.
6. Apakah titik yang tepilih merupakan titik tujuan? Jika ya, maka kumpulan titik terpilih atau *predecessor* merupakan rangkaian yang menunjukkan lintasan terpendek.
7. Begitu seterusnya sampai semua titik terpilih

Pseudocode Algoritma *Dijkstra*

Berikut adalah *pseudocode* algoritma *dijkstra* yang diperlukan untuk melakukan proses pencarian jalur terpendek :

```
Procedure NavMeshAgent
{
        Library untuk memanggil djikstra dan mengetahui posisi koordinat.
}
Procedure LineRenderer
{
        Library untuk mengetahui garis tujuan dan warna.
}
Estimasi jarak : float
Temp_est_jarak: float

Procedure dijkstra()
{
        Mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya.
        Masukan: matriks ketetanggaan (m) dari graph berbobot G dan simpul
        awal a
        Keluaran: lintasan terpendek dari a ke semua simpul lainnya
}
Deskripsi
SET nav NavMeshAgent
SET me_line LineRenderer

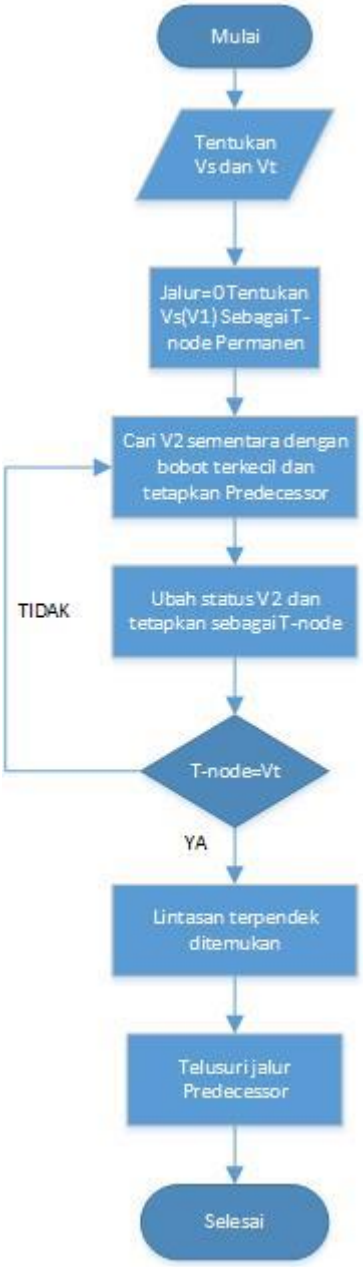
BEGIN
nav.path.ClearCorners ()
nav.destination ← tgt.position
nav.autoRepath ← true
me_line.SetVertexCount (0);
me_line.material ← new Material (Shader.Find ("Sprites/Default"))
        BEGIN
        color ← Color.red
        END

me_line.SetWidth (1f, 1f)
me_line.SetColors (Color.red, Color.red)
path ← nav.path
me_line.SetVertexCount (path.corners.Length)
For x←0 to path.corners.Length do
        BEGIN
        me_line.setPosition(x, path.corners[x])
                IF X >0
                BEGIN
                estimasi_jarak ← vector3.Ditance(path.corners[x-1],
                path.corners[x])
                ENDIF
        ENDFOR
IF estimasi_jarak>0f
```

```
        BEGIN
        temp_est_jarak ← estimasi_jarak
        ENDIF
estimasi_jarak ← 0
END
```

Flowchart Algoritma *Dijkstra*

Berikut ini adalah *flowchart* algoritma *dijkstra* untuk melakukan proses pencarian jalur terpendek



Gambar 2.4 *Flowchart* Algoritma *Dijkstra*

Kelebihan Algoritma *Dijkstra*

Kelebihan dari algoritma *dijkstra* diantaranya yaitu :

1. Algoritma *Dijkstra* dapat menentukan jalur tercepat dengan waktu yang lebih cepat dibandingkan algoritma lainnya.
2. Menggunakan Algoritma *Dijkstra* mempermudah kita dalam mengetahui jarak atau lintasan terpendek dari suatu titik tertentu ke semua titik yang lain.
3. Menggunakan Algoritma *Dijkstra* dalam penerapan di dalam sistem geografis akan menampilakan visualisasi data dalam bentuk peta.
4. Pada penampilan rute atau peta Algoritma *Dijkstra* lebih mudah di baca dan di pahami.
5. Pada rute atau peta dan lintasannya dapat diberikan warna, sehingga penampilan Algoritma *Dijkstra* lebih menarik dan lebih mudah untuk membedakan dari suatu titik tertentu ke titik yang lain.

2.2.4.6 Kesimpulan

Greedy	Hill Climbing	Dijkstra
1. Prinsip pencarian lintasan terpendek memakai fungsi ”Seleksi” dan itu berguna untuk menentukan jalan tersingkat untuk menuju suatu tempat. 2. Hasil analisis berdasarkan bobot-bobot yang berbeda, menunjukkan bahwa semakin banyak bobot yang diberikan, maka semakin akurat pula data yang dihasilkan. Sehingga menghasilkan waktu yang efisien.	5. Ruang masalah dapat didefinisikan sebagai *initial state* (keadaaan awal) menuju keadaan *goal state* (tujuan) dengan ketentuan perulangan dan percabangan sesuai aturan *Hill Climbing.* 6. Aturan produksinya dapat mendefinisikan state ke state lain beserta perubahannya. 7. Menghasilkan solusi evaluasi kedaan yang lebih baik dari keadaan semula. 8. Kombinasi dari metode depth-first search dan metode breadth-first search dengan mengambil kelebihan dari kedua metode tersebut.	6. Algoritma Dijkstra dapat menentukan jalur tercepat dengan waktu yang lebih cepat dibandingkan algoritma lainnya. Karena aturan produksinya menyeleksi node sampai lebih dari satu tahap, sehingga jarak terpendek pertama tidak digunakan sebagai acuan utama. 7. Menggunakan Algoritma Dijkstra dalam penerapan di dalam sistem geografis akan menampilakan visualisasi data dalam bentuk peta yang optimal. 8. Pada penampilan rute atau peta Algoritma Dijkstra lebih mudah di baca dan di pahami.

Dalam penelitian ini menggunakan *Dijkstra* untuk mempermudah mengetahui rute terdekat dari *mapping* denah yang kompleks, sehingga dijkstra merupakan pilihan tepat dilihat dari aturan produksinya yang tidak hanya melihat pada node pertama berikutnya tetapi lebih menekankan kepada jarak optimal pada node kedua berikutnya. Kemudian hal itu sangat efektif dalam menentukan jarak terpendek sesuai dengan tujuan penelitian, dengan pencarian rute yang semakin dekat maka akan memudahkan pengunjung dalam mengakses lokasi yang diinginkan. Lebih dari itu informasi yang dihasilkan akan sangat membantu penyampaian informasi yang mudah didapatkan.

2.2.5 Pemrograman Berorientasi Objek

Object Oriented Programming (OOP) atau Pemrograman Berorientasi Objek (PBO) adalah konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraksi/*blueprint/prototype* dari suatu objek. Sedangkan sistem berorientasi objek merupakan sebuat sistem yang dibangun berdasarkan metode berorientasi objek. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dari komponen lainnya, serta dapat berinteraksi satu sama lain.

Membangun sebuah sistem berorientasi objek harus digunakan juga Pemprograman berorientasi objek (PBO) dapat menggunakan bahasa pemrograman berorientasi objek seperti Java, C#, C++, Smalltalk, Eiffel dan PHP. Ada banyak cara untuk mengabstraksikan dan memodelkan objek – objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Pada saat mengabstraksikan dan memodelkan objek, data, dan proses – proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan[7].

2.2.5.1 Kelas (*Class*)

Kelas adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (method) yang dimiliki oleh objek tersebut . Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode), hubungan (*relationship*) dan arti. Suatu kelas

dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

2.2.5.2 Objek (*Object*)

Objek merupakan instans (perwujudan) dari suatu kelas. Maksudnya, objek merupakan peruwujudan dalam bentuk benda, baik berupa benda nyata, maupun konsep, sesuai dengan deskripsi yang telah didefinisikan oleh kelas. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan dimanipulasi dan dihancurkan.

2.2.5.3 Metode (*Method*)

Metode adalah prosedur atau fungsi yang dimiliki oleh sebuah objek. Sebuah kelas boleh memiliki lebih dari satu metode yang berfungsi untuk memanipulasi objek itu sendiri. Metode juga merupakan cara objek untuk berkomunikasi antar objek.

2.2.5.4 Atribut (*Attribute*)

Atribut adalah sifat karakteristik atau kondisi yang dimiliki oleh suatu objek. Atribut dapat berupa nilai atau elemen–elemen data yang dimiliki oleh objek dalam kelas objek. Contoh manusia memiliki atribut status, berat badan, dan tinggi badan. Di dalam sebuah kelas atribut disebut variabel.

2.2.5.5 Enkapsulasi (*Encapsulation*)

Enkapsulasi ialah lapisan pelindung yang mencegah kode dan data yang secara acak diakses oleh kode lain yang didefinisikan di luar kelas. Oleh karena itu dibutuhkan acces modifier dalam penggunaannya, yaitu *public, private*, dan *protected*.

2.2.5.6 Pewarisan (*Inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya. Konep pemograman dimana sebuah *class* dapat menurunkan *property* dan *metode* yang dimilikinya kepada *class* lain.

2.2.5.7 Antarmuka (*Interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

2.2.5.8 Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor dan kereta.

2.2.5.9 Polimorfisme (*Polymorfism*)

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.2.6 (UML) *Unified Modeling Language*

Unified Modeling Language (UML) merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar spesifikasi UML dijadikan standar *defacto* oleh OMG (*Object Management Group*) pada tahun 1997. UML yang berorientasikan obyek mempunyai beberapa notasi standar. Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. UML dirintis oleh Grady Booch, James Rumbaugh pada tahun 1994 dan kemudian Ivar Jacobson.

Menurut perintisnya, UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Karena tergolong bahasa visual, UML lebih mendepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa visual lebih dekat ke mental model pikiran, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

Seperti yang telah dipaparkan di atas, UML yang merupakan turunan dan beberapa metode mempunyai kumpulan diagram *grafts* sebagai kombinasi dari konsep pemodelan data (*Entity Relationship Diagram*), pemodelan bisnis (*work flow*), pemodelan obyek, dan pemodelan komponen. Diagram *grafts* tersebut merupakan tampiian dari beberapa level abstraksi yang dapat digunakan secara bersama oleh semua proses pada seluruh *lifecycle* pengembangan perangkat lunak serta pada implementasi kebeberapa teknologi yang berbeda. UML memiliki berbagai jenis diagram, diantarnya adalah:[7]

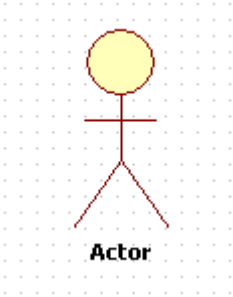
2.2.6.1 *Use case* Diagram

Use case diagram berisi mengenai interaksi antara sekelompok proses dengan sekelompok *actor*, menggambarkan fungsionalitas dari sebuah sistem yang dibangun dan bagaimana sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menangkap kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

Use Case diagram terdiri dari beberapa elemen pemodelan utama, yaitu *Actor, Use Case, Association, Dependency,* dan *Generalization.*

1. *Actor*

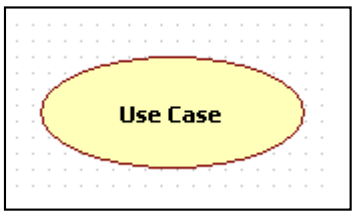
Pada dasarnya *actor* bukanlah bagian dari *use case* diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan stick man. Pada simbol 2.1. menunjukan Simbol *Actor* pada *Use Case* Diagram.



Gambar 2.5 Simbol *Actor* pada *Use Case* diagram

2. *Use Case*

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Pada simbol 2.2. merupakan Simbol *Use Case* pada *Use Case* Diagram.



Gambar 2.6 Contoh *Use Case*

3. *Association*

Associaton menghubungkan *link* antar *element*, dan bukan menggambarkan aliran data / informasi pada sistem. *Association* digunakan untuk menggambarkan bagaimana *actor* terlibat dalam *use case*. Ada 4 jenis relasi yang bisa timbul pada *use case* diagram, yaitu *Association* antara *actor* dan use *case*, *Association* antara *use case*, *Generalization*/*Inheritance* antara *use case*, *Generalization*/*Inheritance* antara *actors*.

4. *Dependency*

   Dependency adalah sebuah *element* bergantung dalam beberapa cara ke *element* lainnya.

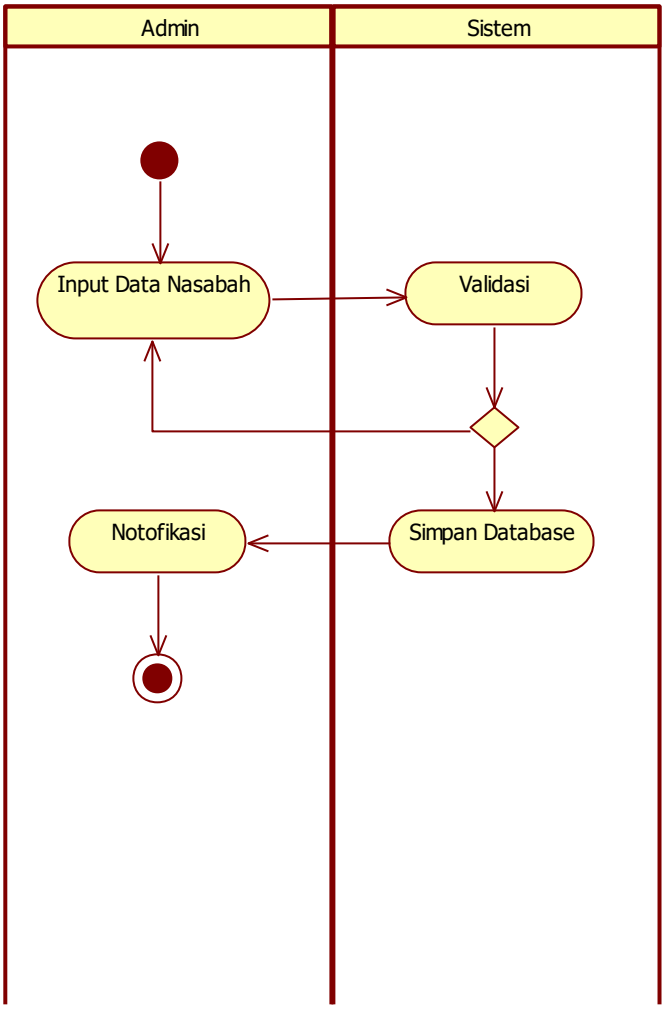
5. *Generalization*

   Generalization disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.

2.2.6.2 *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang diarancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *Action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behavior* internal sebuah sistem.

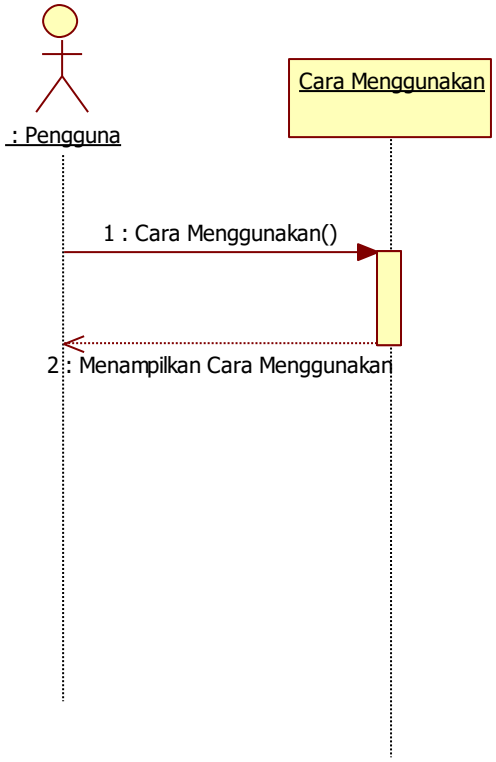


Gambar 2.7 *Contoh* Activity Diagram

2.2.6.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang disusun dalam suatu urutan waktu.

Secara khusus, diagram ini berasosiasi dengan *use case*. *Sequence* diagram menggambarkan *behavior* internal sebuah sistem. Dan lebih menekankan pada penyampaian *message* dengan parameter waktu.



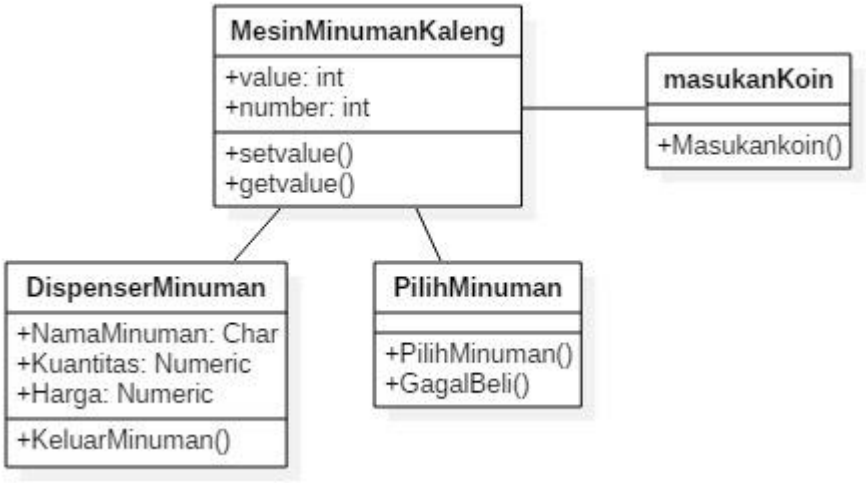
Gambar 2.8 Contoh *Sequence Diagram*

2.2.6.4 *Class* Diagram

Class diagram adalah diagram yang menunjukan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika dan menggambarkan struktur statis dari sebuah sistem[15]. *Class* diagram digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang digunakan.

Class diagram memberikan gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan yaitu bagian

atas adalah bagian nama dari *class*, bagian tengah mendefinisikan *property* / *atribut class*, bagian akhir mendefinisikan *method-method* dari sebuah *class*. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain atau bagian atau kelompoknya.



Gambar 2.9 Contoh *Class Diagram*

2.2.7 Skala *Likert*

Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian[9]. Dengan skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrument yang dapat berupa pertanyaan[9].

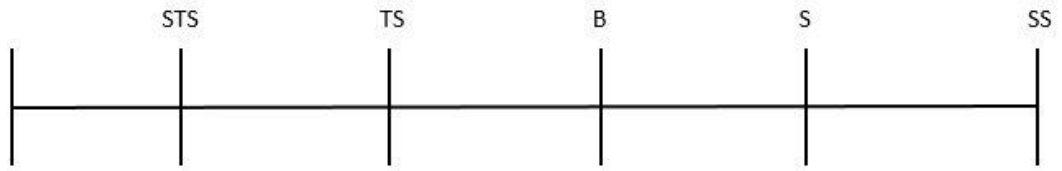
Jawaban setiap item instrument yang menggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negative yang dapat berupa kata-kata antara lain:

1. Sangat 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/hampir tidak pernah/negatif diberi skor 2

5. Sangat Tidak Setuju/tidak pernah diberi skor 1

Cara perhitungan akhir yaitu dengan mencari skor yang diperoleh dari penelitian, sehingga akan didapatkan hasil tingkat persetujuan terhadap penelitian yang dilakukan.

Secara kontinum dapat digambarkan seperti berikut:



Gambar 2.10 Skala Likert

2.2.8 Perangkat Lunak Pendukung

Pada pembangunan peta 3 dimensi di komplek pemerintahan Kabupaten Bandung dibutuhkan perangkat lunak pendukung, diantaranya adalah:

2.2.8.1 Blender 3D

Pada pembangunan peta 3D ini penulis menggunakan Blender 3D untuk melakukan pemodelan tiga dimensi. Blender 3D adalah perangkat lunak sumber terbuka grafika komputer 3D. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan *video*. Blender memiliki beberapa fitur termasuk pemodelan 3D, penteksturan, penyunting gambar bitmap, penulangan, simulasi cairan dan asap, simulasi partikel, animasi, penyunting *video*, pemahat digital, dan *rendering*.

Blender memiliki fitur sama kuat mengatur dalam lingkup dan kedalaman ke ujung lain tinggi 3D *software* seperti *Softimage | XSI, Cinema* 4D, 3ds Max dan Maya. Perangkat lunak ini berisi fitur yang merupakan ciri khas dari model perangkat lunak *high-end*.

Fitur-fitur yang terdapat pada *Blender 3D*:

1. Model: Obyek 3D tipe, termasuk jerat poligon, permukaan NURBS, *Bezier* dan *kurva B-spline*; multiresolusi patung kemampuan; *Modifier stack deformers*; model *Mesh; Python Scripting*
2. *Rigging: Skeleton* kode ciptaan*; Skinning*; lapisan *Bone; B-splines interpolated* tulang
3. Animasi: animasi *editor* non-linear; Vertex framing kunci untuk *morphing*, animasi Karakter berpose editor; deformers animasi, pemutaran Audio; sistem kendala animasi
4. *Rendering: raytracer inbuilt; oversampling, blor* gerak, efek pasca produksi, ladang, non-square *pixel*, lapisan *Render* dan melewati; *Render baking* ke peta UV, Efek termasuk halo, suar lensa, kabut, *vektor motion-blur* proses pasca- , dan proses pasca-*defocus; Ekspor* naskah untuk penyaji eksternal
5. UV *unwrapping: Laurent* dan metode Berdasarkan Sudut *unwrapping*; *unwreapping* berdasarkan jahitan; *falloff* proporsional mengedit peta UV
6. *Shading*: membaur dan *shader specular*; *Node editor*; hamburan Bawah, *shading Tangent*; peta Refleksi
7. Fisika dan Partikel: sistem Partikel dapat dilampirkan ke mesh objek; simulator *Fluida; solver Realtime* tubuh lembut
8. Imaging dan Komposisi: multilayer OpenEXR dukungan; *filter node komposit, konverter,* warna dan operator *vektor*; 8 mendukung prosesor; *sequencer realtime* dekat; Bentuk gelombang dan U / V menyebar plits
9. *Realtime* 3D/*Game* Penciptaan: *editor grafis* logika; *Bullet* Fisika dukungan Perpustakaan; jenis *Shape: polyhedron Convex*, kotak, bola, kerucut, silinder, kapsul, majemuk, dan mesh segitiga statis dengan mode auto penonaktifan; tabrakan Diskrit; Dukungan untuk kendaraan dinamika; Mendukung semua modus pencahayaan *OpenGL; Python scripting*; Audio
10. Lintas *Platfrom* dengan GUI *OpenGL* seragam pada semua *Platfrom* ,siap untuk digunakan untuk semua versi windows (98, NT, 2000, XP), Linux,OS X , *FreeBSD, Irix,* SUN dan berbagai Sistem Oprasi lainnya .

11. Kualitas tinggi arsitektur 3D yeng memungkinkan penciptaan cepat dan efisien.
12. Diekseskusi berukuran kecil, dan distribusi rendah

2.2.8.2 Unity 3D

Unity adalah sebuah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi. Unity bisa untuk games PC dan *games online*. untuk *games online* memerlukan sebuah *plugin*, yaitu *unity web player*, sama halnya dengan *Flash Player* pada *browser*. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari *Javascript*, C#, dan Boo**.**

Pada unity, tidak bisa melakukan desain / *modelling*, dikarenakan unity bukan *tool* untuk mendesain Jadi jika ingin mendesain, harus memerlukan 3D *editor* lain seperti 3dsmax atau Blender. Banyak hal yang bisa di lakukan di unity , ada fitur audio *reverb zone* , *particle effect* , *Sky Box* Untuk menambahkan langit , dan masih banyak lagi tentunya, juga bisa langsung mengedit *texture* dari *editor* seperti *photoshop* dll , unity bagus untuk pemula maupun *expert*.

1. Mendukung 3 bahasa pemrograman, *JavaScript, C#*, dan *Boo*.
2. *Flexible* and *EasyMoving, rotating*, dan *scaling objects* hanya perlu sebaris kode . Begitu juga dengan *Duplicating, removing*, dan *changing properties*.
3. *Multi Platform Game* bisa di *deploy* di PC, Mac, Wii, iPhone, iPad dan *browser*, android.
4. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada *Editor*. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker.*
5. Berbasis .NET –> Penjalanan program dilakukan dengan *Open Source* .NET platform, Mono.

2.2.8.3 Star UML

Star UML adalah software permodelan yang mendukung UML (*Unified Modeling Language*). Berdasarkan pada UML *version* 1.4 dan dilengkapi 11

macam diagram yang berbeda, mendukung notasi UML 2.0 dan juga mendukung pendekatan MDA (*Model Driven Architecture*) dengan dukungan konsep UML. Star UML dapat memaksimalkan pruduktivitas dan kualitas dari suatu *software project*.

2.2.8.4 Android

Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. Android juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan Android, dibentuklah *Open Handset Alliance, konsorsium* dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, *Intel, Motorola, Qualcomm, T-Mobile,* dan *Nvidia.*