



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1. *3D Animation*

Dalam buku *3D Animation Essentials*, menurut Beane (2012) animasi 3D telah berkembang dan tidak terpisah dari televisi, film maupun *video games* dan telah menjadi bagian industri yang penting yang awalnya hanya di pandang sebelah mata. Berbagai macam jenis pekerjaan telah menggunakan 3D sebagai alat bantu seperti arsitektur, hukum, forensik dan obat-obatan.

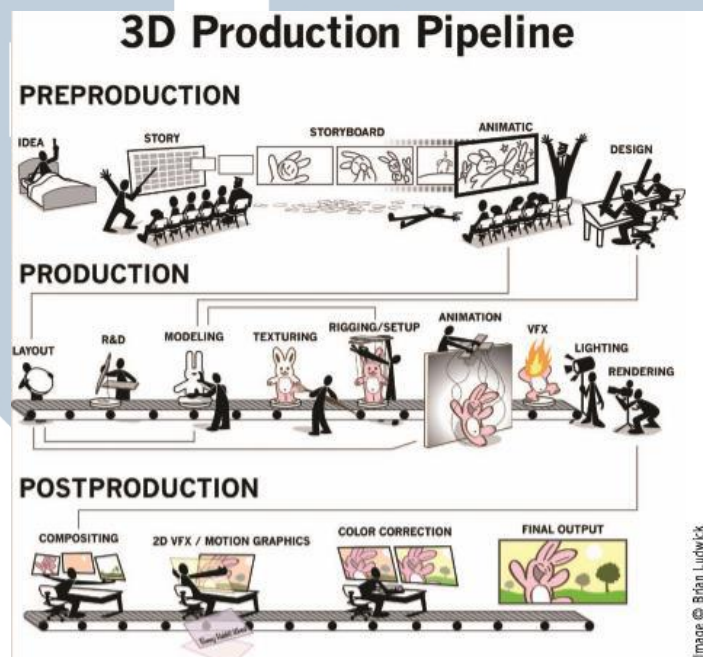
Arti *3D animation* menurut Beane adalah pemahaman umum dari keseluruhan industri dalam cakupan 3D animasi yang menggunakan komputer baik *hardware* maupun *software* dalam tahap produksi. Dalam pembuatan 3D animasi ada bagian yaitu *texturer, rigger, modeler, visual effect* dan *animator*.

2.2. *Pipeline Animation*

Menurut Beane (2012) *pipeline* animasi merupakan sebuah cara untuk mengorganisir kelompok, waktu, *software* dan *hardware* agar dapat mengerjakan suatu dengan teratur untuk membuat 3D animasi ataupun *asset*. *Pipeline* biasanya digunakan oleh studio-studio awal yang masih mencari dana untuk mengerjakan proyek yang akan dikerjakan, sehingga *pipeline* berfungsi sebagai gambaran untuk mengetahui proses pembuatan proyek animasi tersebut.

Pipeline untuk *visual effect, entertainment* dan TV memiliki sedikit perbedaan. Untuk mengerjakan proyek dalam tahanan *pre-production* akan membutuhkan waktu, akan tetapi akan berjalan dengan lancar jika pembuatna

pipeline dibuat dengan *detail* dan jelas. Produk yang mengikuti *pipeline* harus direalisasikan. Industri kreatif ini menghabiskan kebanyakan waktunya pada tahap pre-production untuk melakukan penelitian dan pengembangan akan sesuatu yang studio tersebut akan buat.



Gambar 2.1. *Pipeline Animation*

(*3D Animation Essentials*/Beanne, 2012)

2.3. *Visual Effect*

Dalam buku berjudul *The VES Handbook of Visual Effect*, menurut Zwerman dan Okun (2010) perkembangan *visual effect* dimulai pada tahun 1895 hingga 1905 hasil rekaman yang di dapat oleh kamera sebagai batasan pemberian *visual effect*, termasuk penggantian substitusi *shot* maupun pembagian per *frame*. Film berjudul *The Execution of Mary, Queen of Scots*, *shot* sejarah dari studio Thomas Edison di New Jersey pada tahun 1895 menjadi *visual effect* yang diakui

secara luas. Seseorang yang bernama Alfred Clark bergabung dalam tim studio Thomas Edison mengemukakan teknik dengan cara menghentikan kamera sehingga melakukan pergantian *dummy* yang kepalanya dengan aman dilepas dengan aktor asli yang berpotret sebagai ratu.

Menurut Zwerman dan Okun (2010) penggunaan layar biru mulai digunakan pada tahun 1930s pada film hitam putih, contohnya adalah film berjudul *King Kong* yang menggunakan layar biru tersebut. Untuk mengatasi sensitivitas warna pada film hitam putih, warna biru dapat menghasilkan tidak adanya terkena *exposure*. Sistem lain yang menggunakan selain layar biru adalah sistem yang menggunakan cahaya ultraviolet ataupun emulsi *infrared* yang sensitif pada pergerakan gambar kamera meskipun pada akhirnya ditinggalkan. Teknik yang berhasil digunakan Walt Disney bernama *sodium vapor*. Cara ini menghasilkan matte yang ada dengan cara memaparkan simulasi secara terus melalui sebuah prism pada kemara yang telah di modifikasi menjadi tiga garis kamera.



Gambar 2.2. Kingkong 1930s

(www.renderosity.com)

Menurut Zwerman dan Okun (2010) *visual effect* adalah sebuah kejadian yang tidak bisa diambil secara langsung di dunia nyata pada sebuah film ataupun

media yang ada dengan cara diubah, perumpamaan, ataupun ditingkatkan. *Visual effect* lebih banyak berperan ketika gambar utama selesai di kerjakan. *Live-action capture* dapat ditambahkan ke dalam *visual effect* melalui objek grafik komputer, *matte painting*, lingkungan dan karakter. Ledakan belakangan di peralatan digital memungkinkan beberapa hal yaitu menggabungkan berkas dengan lancar, *computer-generated*, dan *digital sets* menjadi tercapai dan memungkinkan untuk semua tingkatan *visual effect* sebagai hal yang dasar dalam segi alat-alat pembuat pergerakan gambar (hlm.2)

2.4. Non-Photorealistic Rendering

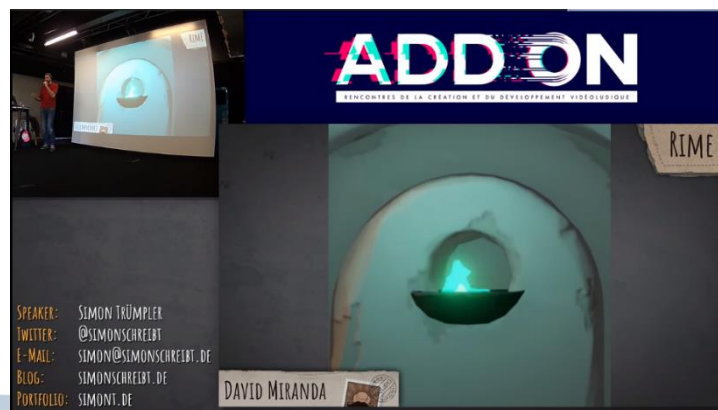
Berdasarkan buku yang berjudul *State of the Art Non-photorealistic Techniques*, Derby dan Lever (2006) mengatakan secara umum teknik-teknik yang digunakan dan syarat pengurangan parameter dalam aplikasi yang digunakan adalah suatu cara yang dapat dilakukan untuk menghasilkan *non-photorealistic rendering*. Untuk membuat suatu hal terlihat *non-photorealistic* harus dilakukan tanpa adanya hal yang pasti dalam tingkat besar atau kecilnya *voxel* di aplikasi. Perubahan ataupun penambahan dapat dilakukan dengan dua cara yaitu mengubah warna-warna yang telah ada ataupun yang sudah diidentifikasi dan menambah tingkat *opacity*. Mendeteksi fitur tersebut dapat melalui sudut pandang dan *gradient*. Meskipun dalam proses *non-photorealistic rendering* dilakukannya penambahan, tidak ada kerangka pasti dalam membuat *non-photorealistic* (hlm.5-7).

Berdasarkan buku yang berjudul *Non-Photorealistic Rendering - A Critical Examination and Proposed System*, menurut Schofield (1994) pengenalan *non-*

photorealistic dalam tahap awal adalah *randomness* dan *noise* yang dijadikan gambar di layar komputer. Sehingga menjadikan tujuan yang diinginkan dalam hal pembuatan *non-photorealistic rendering* (hlm.29).

2.5. *Stylize Visual Effect*

Berdasarkan seorang ahli dalam hal *visual effect* dalam katagori *stylize visual effect* bernama Schreibt (2017) hal yang paling dasar dalam hal *stylize visual effect* adalah distorsi. Salah satu ciri-ciri dari *stylize visual effect* adalah tampilan visualnya salah satunya adalah seperti lukisan dan pergerakan berbeda dengan *realistic visual effect*. Hal lain yang membedakan *visual effect* dengan *stylize* dengan *visual effect realistic* adalah komposisi warna dari *visual effect stylize* tidak sekompleks dengan warna *visual effect* realis.



Gambar 2.3. *Stylize Visual Effect*

(Simon Scheribt, www.youtube.com)

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.6. 3D Visual Effect

Menurut Zwerman dan Okun (2010) 3D *visual effect* adalah efek-efek yang dihasilkan melalui komputer dan disatukan dengan cara *compositing* dari objek-objek komputer, karakter dan lingkungan dengan berbagai cara yang ada untuk dijadikan suatu kesatuan. Salah satu contohnya adalah *visual effect* ledakan yang tidak langsung diambil di dunia nyata, akan tetapi sepenuhnya dibuat melalui komputer.

2.6.1. Shaders

Beanne (2012) berpendapat bahwa instruksi terhadap *software* yang digunakan pada saat *rendering*, memperbolehkan sang *artist* untuk berkreasi dan menentukan tampilan akhir pada permukaan objek yang dipilih pada tahanan *render* akhir merupakan peranan *shader*. Ada beberapa atribut *shader* dalam *maya*, yaitu:

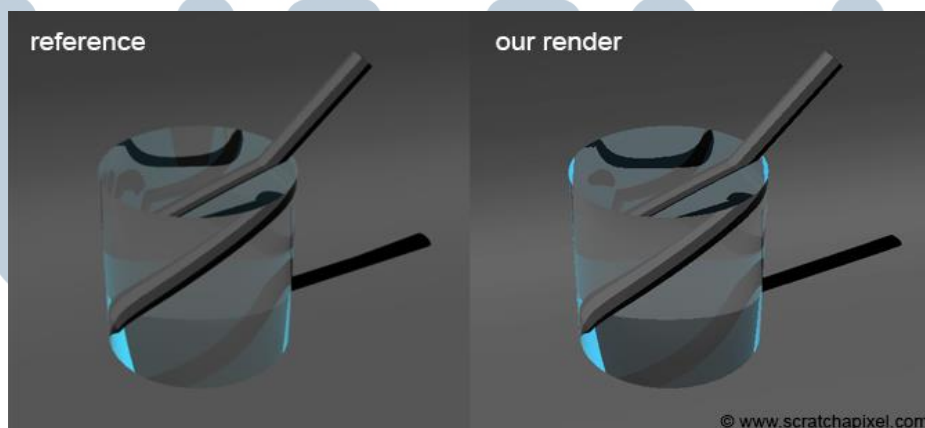
- *Color*, merupakan warna objek tersebut seperti merah ataupun biru, bisa warna dasar saja ataupun dijadikan tekstur.
- *Ambience* adalah simulasi dari pencahayaan di lingkungan sekitar objek ke permukaan objek. Banyak atau sedikitnya warna *ambient* akan mempengaruhi keseluruhan permukaan objek yang terkena *ambient*.
- *Transparency*, merupakan penentuan tembus pandang tidaknya objek tersebut jika terlihat oleh mata.
- *Reflectivity*, akan menentukan seberapa besar *effect* reflektif yang akan ditimbulkan objek.



Gambar.2.4. *Reflectivity*

(www.medium.com)

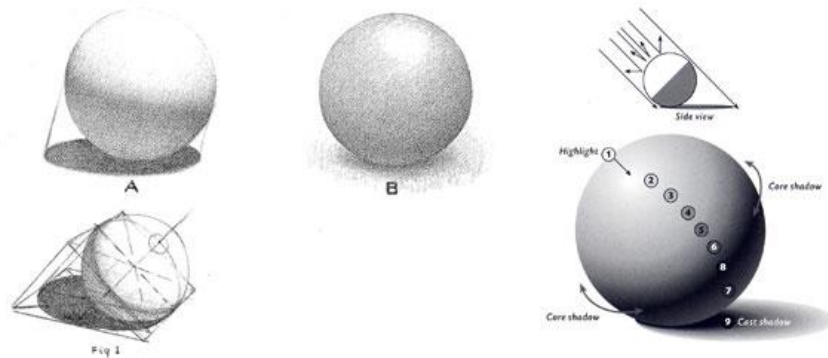
- *Refraction*, perubahan dalam hal arah pencahayaan dari segi kecepatan yang merupakan pencahayaan yang tembus dari permukaan objek yang transparan.



Gambar 2.5. *Refraction*

(www.scratchapixel.com)

- *Transluceny* merupakan jumlah pencahayaan kepada kertas yang dapat di tembus oleh cahaya.
- *Incandescene*, tingkatan dalam kualitas dalam hal *self-illumination*.
- *Specular Highlights* merupakan titik paling terang di antara keseluruhan permukaan objek. Sebagai contoh dalam dunia nyata adalah titik terang diibaratkan dengan sumber cahaya ke arah objek.

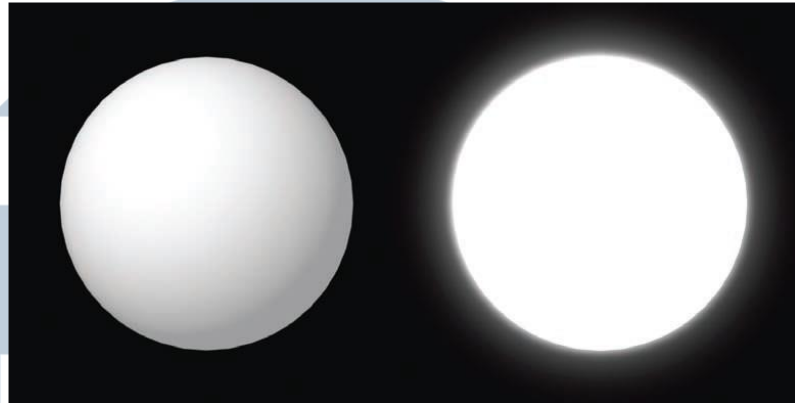


Gambar 2.6. *Specular*

(www.huevaluechroma.com)

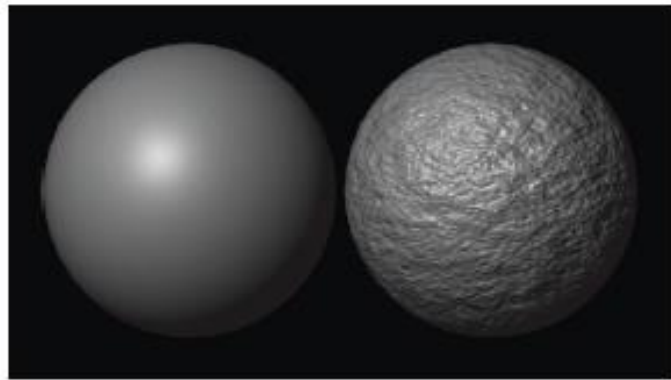
- *Glow* merupakan atribut dalam hal 3D dengan tingkatan standar yang digunakan untuk menambah *effect self-ilumination*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.7. *Incandescence* dan *Glow*
(*3D Animation Essentials*/Beanne, 2012)

- *Bump* digunakan untuk memberi kesan tekstur pada objek yang di berikan *effect* tersebut sehingga menambah pencahayaan dan bayangan pada objek tersebut seperti pada gambar 2.8.



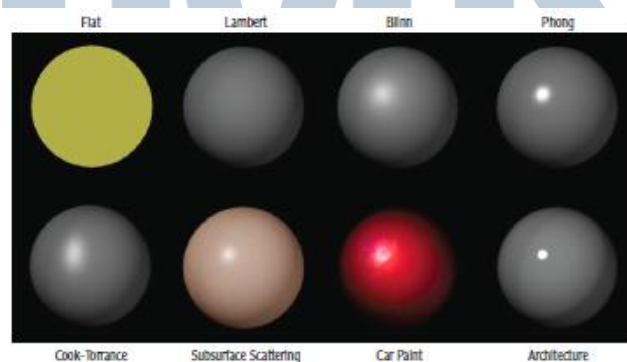
Gambar 2.8. *Bump Material*

(*3D Animation Essentials*/Beanne, 2012)

Dalam *maya*, ada berbagai tipe *shaders* seperti:

- *Flat shaders* merupakan warna 2D yang datar bahkan kepada objek 3D yang di terapkan.

- *Lambert shaders* akan memberikan atau menyediakan *self-illumination* dasar pada objek yang akan diberikan *lambert*.
- *Blinn shaders* akan memberikan *effect* refleksi dan *specular highlights*.
- *Phong shaders* akan memberikan *effect* tampilan yang berkilau dengan penambahan dari segi *specular highlight*.
- *Cook-Torrance shaders* dalam hal *specular highlights* di permukaan objek yang di terapkan akan ditiru oleh tipe *shader* ini.
- *Subsurface scattering shaders* merupakan tipe *shaders* dalam segi perhitungan cahaya yang akan menembus dari permukaan objek.
- *Car paint shaders* merupakan tipe *shaders* yang lebih kompleks dikarenakan terdiri dari dua atau lebih permukaan yang akan berinteraksi satu sama lain.
- *Physical architecture shaders* merupakan simulasi yang dilakukan kepada dunia asli iluminasi untuk membuat berbagai macam objek seperti kaca, keramik, air, besi dan plastik.

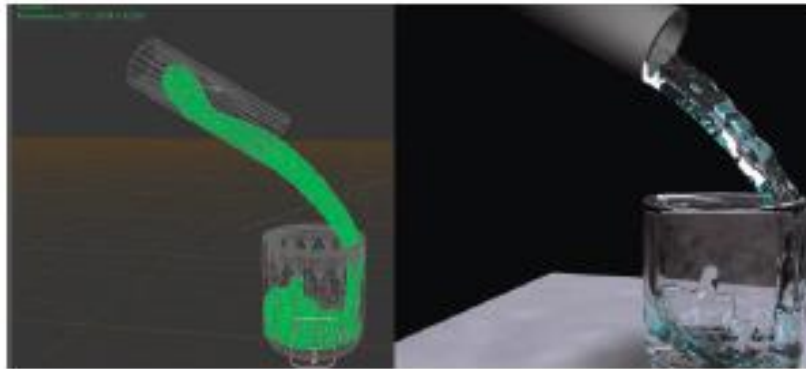


Gambar 2.9. Tipe-Tipe Shader

(3D Animation Essentials/Beanne, 2012)

2.6.2. Fluid

Dalam buku *Fluid Simulation*, menurut Bridson dan Muller (2007) kekuatan pertama dari *fluid* adalah tekanan. Area bertekanan rendah akan di dorong oleh area bertekanan tinggi. Hal yang terpenting adalah ketidakseimbangannya dalam hal tekanan, semakin tinggi tekanannya maka akan menghasilkan tujuan ke arah yang bertekanan rendah seperti pada gambar 2.10.

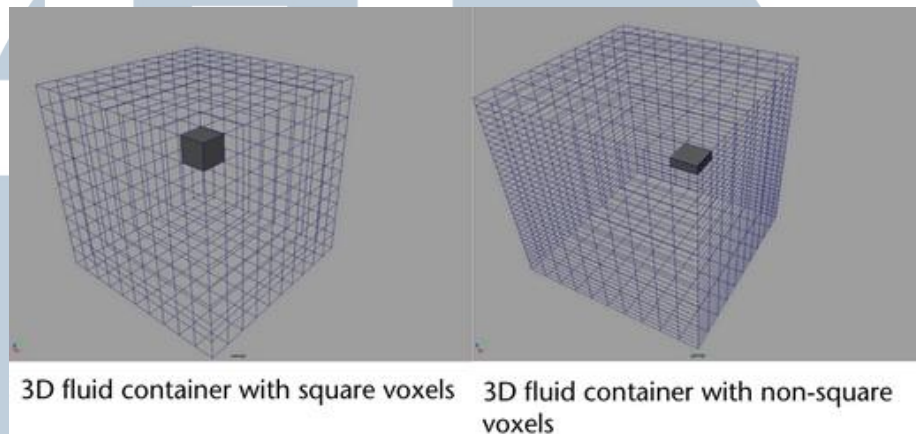


Gambar 2.10. Fluid

(3D Animation Essentials/Beanne, 2012)

Dalam buku *Mastering Autodesk Maya 2016*, menurut Palamar (2016) *fluids* sering dibaratkan dengan permukaan yang tidak tetap atau berubah-ubah. *Emitter* dalam *Maya Fluids* dapat menghasilkan berbagai macam efek seperti api, ledakan, asap, awan, dan seterusnya. *Fluid containers* dapat di pelajari dari *scene* di *maya* dalam pemikiran cuplikan kecil. Penggunaan maksimal dapat digunakan ketika membuat api, ledakan, dan asap. Gaya dinamik akan muncul dalam sebuah batasan untuk menghasilkan efek tersebut. 3D dan 2D merupakan batasan yang digunakan dalam membuat efek. Dalam 3D mengutamakan dalam hal volume

sedangkan 2D lebih ke arah panel datar yang lebih cepat di kalkulasikan (hlm.677).



Gambar 2.11. *3D Container*

(www.help.autodesk.com)

Penggunaan *fluid* dilakukan ketika penggunaan partikel ditarafkan berjumlah sangat banyak seperti 100.000 partikel. Ketika kebutuhan dengan jumlah partikel sebanyak tersebut disarankan untuk menggunakan *fluid*. Partikel dan *fluid* dapat berkerja sama satu sama lain. Evaluasi di *fluid* dilakukan berdasarkan *frame* yang sebelumnya sehingga disarankan untuk melakukan pemilihan *playback* untuk *fluid* mengkalkulasikan secara benar. Ada cara lain untuk menghasilkan *fluid* dalam batasan *fluid* yaitu menambahkan *fluid emitter*. Perbedaan dari partikel dan *fluids* adalah *fluid* yang dihasilkan di oleh *fluid emitter* hanya akan muncul dalam batasan *fluid*.

Fluid emitter memiliki berbagai macam tipe, seperti:

- *Volume emitter* akan menghasilkan *fluid* didalam area yang sudah di identifikasi dalam bentuk ruang yang primitif seperti kubus maupun bola.

- *Omni emitter* merupakan sebuah titik yang akan menghasilkan partikel keseluruhan arah.
- *Curve emitter* akan menghasilkan *fluids* bersama dengan ukuran panjang belokan *NURBS*.
- *Surface emitter* merupakan *emitter* yang akan menghasilkan *fluid* dari permukaan objek 3D yang dipilih.

Turbulence, Drag, Volume Axis, dan Vortex merupakan bagian dari *field* dinamik yang dapat mengontrol *fluid*. Dalam penambahan konten seperti *velocity, temperature, density* atau *fuel* di dalam *3D container* akan ada tiga pilihan seperti:

- Yang pertama adalah *Dynamic Grid*, jika menggunakan opsi ini pengguna akan mengetahui bahwa *value* akan berubah selama waktu berlangsung sebagai hasil dari simulasi, hal ini juga berpengaruh terhadap element *fluid*.
- Kedua adalah *Static Grid*, opsi ini merupakan kebalikan dari *dynamic grid*.

Simulasi tidak akan merubah *value* yang dibuat untuk elemen-elemen tersebut. Sebagai contohnya jika ingin membuat efek awan disarankan menggunakan opsi ini.

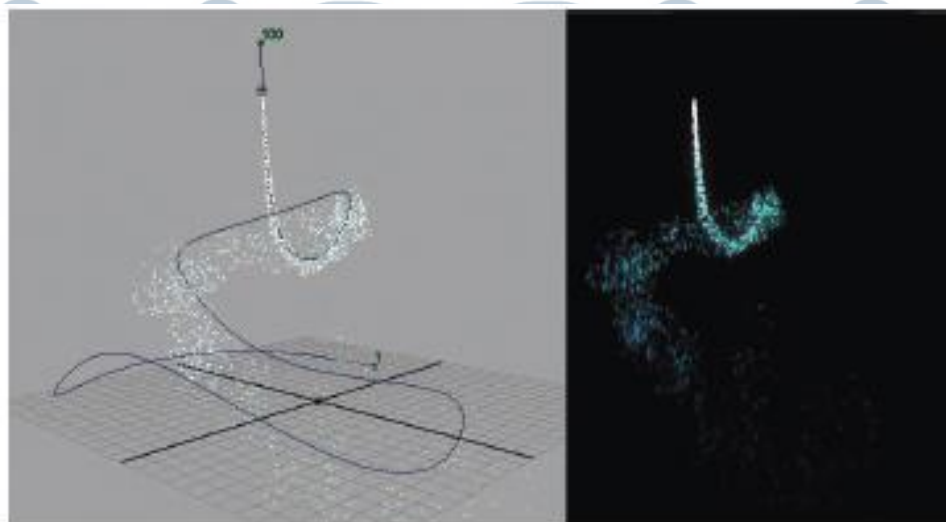
- Ketiga adalah *Gradient*, opsi ini menggunakan *value* antara 0 dan 1. *Value* akan berdampak pada simulasi akan tetapi tidak akan berubah karena hal tersebut.

Simulasi yang dilakukan dalam *fluid* dapat dirender menggunakan *maya software* ataupun *mental ray*. *Fluid* mempunyai *incandescent value* yang dapat digunakan untuk menghasilkan objek cahaya ketika dalam tahap *final gathering* di *render*. Bagian cahaya jika menggunakan *real light*, maka objek-objek selain *fluid*

akan menghasilkan bayangan juga. Jika menggunakan *real-shadow casting lights*, pastikan tidak mencentang *self shadow* untuk kalkulasi ketika *render* tidak mengkalkulasikan bayangan secara dua kali. *Auto Resize* dapat menghasilkan dampak batasan *fluid* untuk berubah menyesuaikan ukuran *fluid* yang disimulasikan. *Density* berperan dalam kalkulasi untuk perenggangan atau besar kecilnya efek *auto resize*.

2.6.3. Particle

Menurut Palamar (2016) partikel adalah simulasi dari gaya yang bersifat dinamik dalam sebuah ruang di *Maya*. Partikel-partikel tersebut dapat berperan dalam membuat efek yang dianimasikan yang susah dicapai dengan cara penggunaan *keyframe* standar. Partikel dapat dipengaruhi dengan cara penambahan *field* yang dapat membuat partikel bertambakan ataupun jatuh, mengambang, dan sebagainya seperti pada gambar 2.12.



Gambar 2.12. *Particles*

(*3D Animation Essentials* 2016/Benny, 2012)

Dalam buku berjudul *Learning Autodesk Maya 2008*, menurut Autodesk Agency (2008) penggunaan partikel biasa digunakan dalam hal mensimulasi fenomena alam yang kompleks. Pergerakan secara acak dan kompleks dari beberapa komponen-komponen individual seperti percikan, api, salju, hujan, asap, dan debu. Sumber untuk mengeluarkan partikel ke angkasa disebut *emitter*. Beberapa bentuk emiter yang dapat digunakan antara lain *directional*, *volume*, *surface*, *omni*, *per-point*, *curve*, dan *texture* (hlm.372).

Ada beberapa hal yang harus diketahui dalam hal partikel yaitu

- *Force*, kuantitas yang dihasilkan yang dapat berasal dari *fields*, *springs*, dan *expressions*.
- *Mass*, sebuah atribut yang sudah ada dari dasar sebuah objek partikel yang ada.
- *Scalar*, kuantitas dalam hal angka dalam komponen yang spesifik. Berat dan waktu adalah contoh yang tepat dalam hal skala.
- *Vector*, kuantitas dalam hal arah dan *magnitude*.
- *Position (vector)*, lokasi partikel dalam dunia tersebut.
- *Velocity (vector)*, perubahan dalam hal posisi yang berubah dari waktu ke waktu.
- *Speed (scalar)*, tingkatan dalam hal partikel yang tanpa arah.
- *Acceleration (vector)*, perubahan dalam hal *velocity* dari waktu ke waktu (hlm.432-433).

Hal lain yang harus diketahui adalah *creation* and *runtime expressions*. Dari segi *creation expression*, evaluasi hanya akan dilakukan sekali pada setiap partikel yang ada pada objek partikel ketika partikel mulai muncul dari *emitter*. Sedangkan *runtime expression*, evaluasi akan dilakukan setiap *frame* yang ada setidaknya sekali akan tetapi tidak pada awal munculnya partikel tersebut. Ada dua tipe *runtime expression* yang ada yaitu *runtime* sebelum *dynamic* dan *runtime* setelah *dynamic*. Kedua hal tersebut dapat dilakukan di sesi bagian *particleShape* (hlm.434).

2.7. *Fire Sparks*

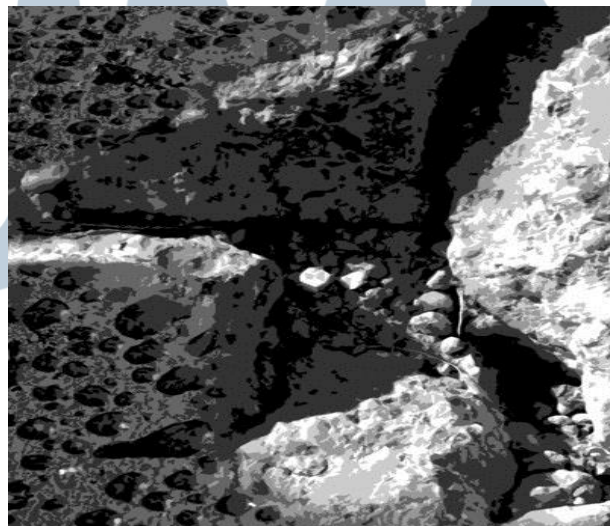
Dalam buku berjudul *Pyrotecnic Spark Generation*, menurut Kenneth dan Bonnie (1999) durasi berlangsungnya *sparks* dapat di ukur dari hasil pengurangan karbon kepada oksigen dari terkenanya eksposur partikel-partikel yang berada dalam atmosfer. Ketika oksigen dalam atmosfer menjadi larut dalam *sparks*, partikel karbon masih dapat dicapai oleh oksigen dan terkubur didalamnya. Proses pembakaran produk dapat mencapai hasil *sparks* yang diinginkan, dengan semakin lamanya *sparks* berlangsung di udara dan tampilan densitasnya.

Kenneth dan Bonnie (1999) mengatakan mengubah tipe dan banyaknya pembakaran merupakan kemampuan yang dapat menghasilkan sebuah kesempatan dalam pembuatan *sparks* untuk mengontrol durasi *sparks* dan warna yang ditampilkan pada *sparks* tersebut. Dalam formula seandainya pada suatu benda memiliki persentase cairan yang tinggi, partikel yang dihasilkan akan berwarna lebih merah, hal tersebut dikarenakan benda tersebut lebih terlindungi dari oksidasi udara sekitar. Ukuran *sparks* yang terbentuk menentukan jangka

panjangnya *sparks* tersebut hidup meskipun hal tersebut belum di laporkan. Hal tersebut dapat terjadi dikarenakan dua hal yaitu reaksi yang berhubungan antara pengurangan energi dan oksigen atmosfer. Benda yang terbentuk dari bahan dasar sodium dapat membuat *sparks* berwarna kuning (hlm.55-56).

2.8. Demolition Dust

Berdasarkan artikel berjudul *Demolition Dust Hazards and Control*, menurut Bosstek (2018) debu merupakan arti paling dasar dari pengertian partikel yang berbentuk. Ukuran partikel tersebut bervariasi, ukuran terkecil dari partikel tersebut adalah 500 micro dalam satuan diameter dari seluruh aspek benda padat. Ukuran partikel terbentuk dalam berbagai ukuran, semakin kecil maka semakin ringan begitu pula sebaliknya semakin besar maka akan semakin berat. Pembuatan dan perobohan bangunan dapat menciptakan berterbangannya debu di sekitar bangunan tersebut.



Gambar 2.13. Permukaan Pecahan Bangunan Yang Dapat Menjadi Debu

(www.bosstek.com)

2.9. Dust

Menurut Palamar (2010) partikel dikalkulasikan dengan adanya pengaruh partikel dengan dunia sekitar mereka. Simulasi partikel tidak terpengaruh dengan berbagai faktor atmosfer seperti densitas udara, tekanan, dan suhu. Partikel sebenarnya dapat bergerak tanpa adanya batasan-batasan yang ada di dunia maupun angkasa. (hlm.154).

Berdasarkan artikel yang berjudul *Airborne Dust Particles*, menurut Girts (2016) segala jenis tanah yang dapat bergerak di udara dapat berasal dari berbagai tempat seperti industri dan individual. Ada beberapa contoh debu yang dapat ditemukan di lingkungan sekitar seperti :

- Debu mineral, material-material yang mengandung *crystalline silica*, karang, dan debu semen.



Gambar 2.14. Debu Mineral

(imimg.com)

- Debu metalik, seperti *lead*, nikel, *cadmium*, dan debu *beryllium*.

- Debu kimia, seperti pestisida dan kimia gabungan.
- Tanaman dan organik, seperti kayu, tepung, polen, dan debu teh.
- *Biohazards*, seperti spora, partikel yang aktif bergerak, dan cetakan.

Sumber debu dapat bersumber dari berbagai hal, seperti:

- Lahan kosong atau lahan terbuka



Gambar 2.15. *Vacant Land*

(www.billgemmill.com)

- Kegiatan pertambangan dan konstruksi



Gambar 2.16. Kegiatan Pertambangan

(www.bone.go.id)

- Kegiatan *maintence*
- Tempat-tempat yang dapat berapi seperti tempat api, hutan dan perkemahan.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA