KOMPUTER GRAFIK

Penyusun

Achmad Udin Zailani

Hiram Sulistio Sibarani

Agung Perdananto

UNPAM PRESS
Lembaga Penerbit & Publikasi

Jl. Surya Kencana No. 1 Pamulang

Gd. A, Ruang 212 Universitas Pamulang

Tangerang Selatan – Banten

KOMPUTER GRAFIK

Penulis:
Achmad Udin Zailani
Hiram Sulistio Sibarani
Agung Perdananto

ISBN: 978-623-6352-28-1

Editor:
Nurjaya

Desain sampul:
Tri Anggoro Seto

Tata Letak:
Ramdani Putra

Penerbit:
Unpam Press

Redaksi:
Jl. Surya Kecana No. 1
Pamulang – Tangerang Selatan
Telp. 021-7412566
Fax. 021 74709855
Email: unpampress@unpam.ac.id

Cetakan pertama, 9 Agustus 2021
Hak cipta dilindungi undang-undang.
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin penerbit.

DATA PUBLIKASI UNPAM PRESS
| Lembaga Penerbit dan Publikasi Universitas Pamulang

Gedung A. R. 212 Kampus 1 Universitas Pamulang

Jalan Surya Kencana Nomor 1 Pamulang Barat, Tangerang Selatan, Banten

Website: www.unpam.ac.id | Email: unpampress@unpam.ac.id

Komputer Grafik/ Achmad Udin Zailani, Hiram Sulistio Sibarani, Agung Perdananto-1STed

ISBN. 978-623-6352-28-1

1. Komputer Grafik I. Achmad Udin Zailani II. Hiram Sulistio Sibarani III. Agung Perdananto

M165-09082021-01

Ketua Unpam Press: Pranoto

Koordinator Editorial: Aden, Ali Madinsyah

Koordinator Hak Cipta: Susanto

Koordinator Produksi: Dameis Surya Anggara

Koordinator Publikasi dan Dokumentasi: Kusworo

Desain Cover: Putut Said Permana

Cetakan pertama, 9 Agustus 2021

Hak cipta dilindungi undang-undang.

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin penerbit.

MATA KULIAH
KOMPUTER GRAFIK

IDENTITAS MATA KULIAH

Program Studi	:	KOMPUTER GRAFIK
Sks	:	3 Sks
Prasyarat	:	-
Semester	:	IV
Deskripsi Mata Kuliah	:	Mata kuliah ini merupakan mata kuliah wajib program studi Teknik Informatika S-1 dengan materi yang dibahas: Pengenalan Grafika Komputer, Titik dan Palet Warna, Algoritma Garis Dan Poligon, Algoritma Lingkaran dan Elips, Algoritma Pembuatan Kurva, Algoritma Pewarnaan Bidang, Proyeksi Geometri Bidang, Transformasi, Translasi dan Rotasi, Dilatasi, Viewing dan Clipping, Antialiasing, Grafika 3 Dimensi, Mampu Mengunakan Api Untuk Pemrograman Komputer Grafik, dan Membuat Aplikasi Grafika Komputer.
Capaian Pembelajaran	:	Setelah menyelesaikan mata kuliah ini, mahasiswa mampu membuat aplikasi grafika komputer 2D melalui kode program dengan baik.
Penyusun	:	1. Achmad Udin Zailani, S.Kom., M.Kom. (Ketua) 2. Hiram Sulistio Sibarani, S.Kom., M.Kom. 3. Agung Perdananto, S.Kom., M.Kom

Tangerang Selatan, 12 Maret 2021

Ketua Program Studi Teknik Informatika S-1	Ketua Tim Penyusun
Achmad Udin Zailani, S.Kom., M.Kom.	Achmad Udin Zailani, S.Kom., M.Kom.
NIDN : 0429058303	NIDN : 0429058303

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan hidayat-Nya sehingga modul bahan ajar Komputer Grafik dapat terselesaikan

Modul bahan ajar ini semoga dapat digunakan dimana sering kali dalam kehidupan sehari-hari, baik dalam dunia kerja, dunia bisnis maupun kehidupan pribadi kita tidak lepas dengan persoalan riset operasi. Kita tidak sadar bahwa yang sering kali kita lihat atau kita alami dapat diselesaikan dengan Komputer Grafik.

Sebagai mata kuliah yang diajarkan pada mahasiswa minimal di tahun ke-3 perguruan tinggi, maka Komputer Grafik membutuhkan pengetahuan pendahuluan (*prerequisite*) yang cukup sehingga materi-materi yang disajikan dapat difahami oleh mahasiswa. Selain itu, diperlukan pengetahuan dan kemampuan pemrograman yang kuat, agar konsep, teori, metode maupun algoritma yang dipelajari dapat diterapkan menggunakan bahasa komputer modern sehingga daya output dari perkuliahan ini dapat lebih terlihat nyata.

Modul ini masih belum sempurna dan belum tertata dengan rapi. Bahan-bahan yang melengkapi modul ajar ini masih harus disempurnakan dan ditambahkan. Namun paling tidak, bagi mahasiswa,adanya modul ini dapat menjadi bahan acuan dalam belajar. Bagi dosen, modul ini juga menjadi alat untuk selalu menyegarkan pengetahuannya sekaligus menunjukkan expertise nya dalam dunia atau komunitas ilmu yang relevan. Bagi institusi, tentunya adanya modul ini menjadi aset dan akan memperkuat posisi sebagai institusi pendidikan yang senantiasa menyediakan sarana yang menunjang

Kesuksesan proses pembelajaran di perguruan tinggi. Mata kuliah ***Komputer Grafik*** mempelajari tentang materi ***Pengenalan Grafika Komputer, Titik dan Palet Warna, Algoritma Garis Dan Poligon, Algoritma Lingkaran dan Elips, Algoritma Pembuatan Kurva, Algoritma Pewarnaan Bidang, Proyeksi Geometri Bidang, Transformasi, Translasi dan Rotasi, Dilatasi, Viewing dan Clipping, Antialiasing, Grafika 3 Dimensi, Mampu Mengunakan Api untuk Pemrograman Komputer Grafik, dan Membuat Aplikasi Grafika Komputer***. Buku Ajar ini disusun untuk memudahkan mahasiswa dalam mempelajari dan memahami serta mengaplikasikan materi mata kuliah Komputer Grafik.

Akhirnya penulis sampaikan terimakasih atas semua bantuan dan dukungan dari semua pihak yang tidak disebutkan satu persatudalam proses penulisan. Semoga buku/ modul ini bermanfaat bagi kita semua

Tangerang Selatan, 12 Maret 2021

Tim Penyusun

DAFTAR ISI

KOMPUTER GRAFIK ii
DATA PUBLIKASI UNPAM PRESS iii
MODUL MATA KULIAH iv
KATA PENGANTAR vi
DAFTAR ISI vii
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
PERTEMUAN 1 1
PENGENALAN GRAFIKA KOMPUTER 1
TUJUAN PEMBELAJARAN 1
URAIAN MATERI 1
1. Pengantar Komputer Grafik 1
2. Gambaran Umum Mata Kuliah Komputer Grafik 4
3. Pengertian Pixel 10
SOAL LATIHAN/TUGAS 11
REFERENSI 11
PERTEMUAN 2 12
TITIK DAN PALET WARNA 12
A. TUJUAN PEMBELAJARAN 12
B. URAIAN MATERI 12
1. Membaca Dan Menulis Pixel 12
2. Membuat Palet Warna 13
3. RGB Dan HSV 17
C. SOAL LATIHAN/TUGAS 22
D. REFERENSI 22

PERTEMUAN 3 23
ALGORITMA GARIS DAN POLIGON 23
A. TUJUAN PEMAHAMAN PEMBELAJARAN 23
B. URAIAN MATERI 23
1. Algoritma Pembentukan Garis 23
2. Algoritma Bresenham 37
3. Switch 40
C. SOAL LATIHAN/TUGAS 53
D. REFERENSI 53
PERTEMUAN 4 54
ALGORITMA LINGKARAN DAN ELIPS 54
A. TUJUAN PEMBELAJARAN 54
B. URAIAN MATERI 54
1. Algoritma Pembuatan Lingkaran 54
2. Algoritma Pembentukan Elips 62
C. SOAL LATIHAN/TUGAS 66
D. REFERENSI 66
PERTEMUAN 5 67
ALGORITMA PEMBUATAN KURVA 67
A. TUJUAN PEMBELAJARAN 67
B. URAIAN MATERI 67
1. Kurva 67
2. Algoritma Bezierbezier Alg 68
C. SOAL LATIHAN/TUGAS 81
1. Bagaimana persamaan polinomial untuk kurva bezier dengan 6 titik, 7 titik dan 8 titik kontrol 81

2. Diketahui 6 buah titik kontrol dengan koordinat C0(-45,-15), C1(-40,-12), C2(-32,-2), C3(-24,0), C4(10,12), C5(24,20) (dengan menggunakan Δt=0.065 maka tentukanlah: 81

a. berapa jumlah titik yang harus digambar untuk membentuk kurva bezier 81

b. berapa nilai titik pada saat t=0.55 81

D. REFERENSI 81

PERTEMUAAN 6 82

ALGORITMA PEWARNAAN BIDANG 82

A. TUJUAN PEMBELAJARAN 82

B. URAIAN MATERI 82

1. Pengertian Pewarnaan Bidang 82
2. Algoritma Perwanaan Bidang 83

C. SOAL LATIHAN/TUGAS 95

D. REFERENSI 96

PERTEMUAAN 7 97

PROYEKSI GEOMETRI BIDANG 97

A. TUJUAN PEMBELAJARAN 97

B. URAIAN MATERI 97

1. Pengertian Proyeksi Geometri 97
2. Taksonomi Proyeksi Geometri 98
3. Proyeksi Paralel 99
4. Proyeksi Perspektif 102

C. SOAL LATIHAN/TUGAS 105

D. REFERENSI 105

PERTEMUAN 8 106

TRANSFORMASI, TRANSLASI DAN ROTASI 106

A. TUJUAN PEMBELAJARAN 106

B. URAIAN MATERI 106
1. Pengertian Tranformasi 106
2. Translasi 107
3. Rotasi 109
C. SOAL LATIHAN\TUGAS 112
D. REFERENSI 112
PERTEMUAN 9 113
DILATASI 113
A. TUJUAN PEMBELAJARAN 113
B. URAIAN MATERI 113
1. Pengertian Dilatasi 113
2. Shear 115
C. SOAL LATIHAN/TUGAS 116
D. REFERENSI 116
PERTEMUAN 10 117
VIEWING DAN CLIPPING 117
A. TUJUAN PEMBELAJARAN 117
B. URAIAN MATERI 117
1. Viewing 117
2. Clipping 121
C. SOAL LATIHAN/TUGAS 126
D. REFERENSI 126
PERTEMUAN 11 127
ANTIALIASING 127
A. TUJUAN PEMBELAJARAN 127
B. URAIAN MATERI 127
1. Teknik Antialiasing 127

C. SOAL LATIHAN/TUGAS 138
D. REFERENSI 138
PERTTEMUAN 12 139
GRAFIKA 3 DIMENSI 139
A. TUJUAN PEMAHAMAN PEMBELAJARAN 139
B. URAIAN MATERI 139
1. Konsep Grafika 3D 139
2. Berbagai Proyeksi Dalam Grafika 3D 141
C. SOAL LATIHAN/TUGAS 152
D. REFERENSI 152
PERTEMUAN 13 153
MAMPU MENGUNAKAN API UNTUK PEMROGRAMAN KOMPUTER GRAFIK 153
E. TUJUAN PEMBELAJARAN 153
F. URAIAN MATERI 153
1. Pengertian OpenGL 153
2. Pengenalan Pemrosesan Citra Digital 156
G. SOAL LATIHAN/TUGAS 163
H. REFERENSI 163
PERTEMUAN 14 164
MEMBUAT APLIKASI GRAFIKA KOMPUTER 164
A. TUJUAN PEMBELAJARAN 164
B. URAIAN MATERI 164
1. Aplikasi Grafika komputer 164
2. Bahan UAS 174
C. SOAL LATIHAN/TUGAS 175
D. REFERENSI 175

GLOSARIUM 176

DAFTAR PUSTAKA 177

DAFTAR TABEL

Table 1. 1 Kaitan Ilmu Grafika komputer dengan Ilmu Lain 4

Table 5. 1 Tabel titik kurva Bezier untuk $\Delta t=0.1$ 70

Table 5. 2 Tabel titik kurva Bezier untuk $\Delta t=0.01$ 72

Table 7. 1 Proyeksi Paralel VS Proyeksi Perspektif 103

DAFTAR GAMBAR

Gambar 2. 1 Menampilkan panel swatches 15

Gambar 2. 2 Menu swatches 15

Gambar 2. 3 Menambahkan warna baru pada swatches 16

Gambar 2. 4 Menentukan warna baru pada swatches 16

Gambar 2. 5 Warna yang telah dibuat 17

Gambar 2. 6 Model warna RGB 18

Gambar 2. 7 Warna HSV 21

Gambar 3. 1 Letak titik dalam koordinat cartesius..... 25

Gambar 3. 2 Perbandingan antara Cartesian dan Screen Cordinates..... 25

Gambar 3. 3 Posisi koordinat Z mengarah keatas 26

Gambar 3. 4 Posisi koordinat Y mengarah keatas 26

Gambar 3. 5 Dibutuhkan pembulatan dari nilai **y** yang disepertemuankan letak titik piksel berupa bilangan bulat..... 28

Gambar 3. 6 Letak titik kordinat menggunakann perhitungan brute force algorithm dalam grafik raster. 30

Gambar 3. 7 (a) Gambar garis dengan kemiringan m > 1, terdapat jarak diantara titik (b) posisi setelah dilakukan interpolasi 31

Gambar 3. 8 Gambar raster graphics pemmbentukan garis memakai hasil perhitungan algoritma Digital Differential Analyzer. .....**Error! Bookmark not defined.**

Gambar 3. 9 Gambar garis yang terletak antara titik NE dan E dan memiliki garis tengah M. 37

Gambar 3. 10 Poligon berbentuk sederhana (a) dan poligon bentuk tidak sederhana (b)..... 40

Gambar 3. 11 ilustrasi penggunaan metode scan line 42

Gambar 3. 12 Ilustrasi penggunaan aturan paritas dan pengisian warna.... 42

Gambar 3. 13 ilustrasi penghitungan aturan paritas dalam gambar verteks. 43

Gambar 3. 14 Bentuk poligon dan bagian sis horizontal 44

Gambar 3. 15 Ilustrasi penggunaan Boundary Fill pada raster..... 48

Gambar 3. 16 Ilustrasi bentuk titik tetangga, 4 titik dan 8 titik tetangga 49

Gambar 4. 1 Adanya jarak/gap dikarenakan perubahan gradien sehingga menghasilkan kordinat pada piksel menjadi tidak merata. 55

Gambar 4. 2 Bentuk lingkaran simetris delapan titik 56

Gambar 4. 3 ilustrasi letak garis yang terletak di titik antara (xk+1, yk) juga (xk+1, yk − 1)..... 57

Gambar 4. 4 Ilustrasi letak piksel dengan pembentukan lingkaran yang memiliki titik pusat yaitu (0,0) dengan r sebesar 8..... 62

Gambar 5. 1 Bezier..... 68

Gambar 5. 2 Fungsi kurva Bezier dengan 4 titik..... 69

Gambar 5. 3 Persamaan Polinomial Bernstein 70

Gambar 5. 4 Persamaan kurva Bezier dengan n titik control..... 70

Gambar 5.5 Persamaan polinomial untuk kurva bezier dengan n titik kontrol74

Gambar 5. 6 Segitiga pascal dan persamaan polynomial 75

Gambar 5.7 Persamaan polinomial untuk kurva bezier dengan n titik kontrol79

Gambar 5. 8 Segitiga pascal dan persamaan polynomial 80

Gambar 6. 1 Tiga warna cukup untuk mewarnai graf ini 84

Gambar 6. 2 Pakai algoritma Welch- Powell buat memberi warna graf G ... 85

Gambar 6. 3 Permasalahan isama idengan icontoh i1, iuntuk igraf iH 85

Gambar 6. 4 Graf bipartit K2.4 86

Gambar 6. 5 Graf bipartit 87

Gambar 6. 6 Graf G 87

Gambar 6. 7 Graf bidang 88

Gambar 6. 8 Graf planar (graf bidang) G 89

Gambar 6. 9 Pewarnaan region 90
Gambar 6. 10 Graf bidang M 90
Gambar 6. 11 Pewarnaan simpul..... 91
Gambar 6. 12 Algoritma Welch Powell..... 91
Gambar 6. 13 Algoritma Welch Powell..... 92
Gambar 6. 14 Gambarkan graf 93
Gambar 6. 15 Bilangan kromatik..... 94
Gambar 7. 1 Proyeksi Geometri Bidang..... 97
Gambar 7. 2 Proyeksi dan Bidang Proyeksi..... 98
Gambar 7. 3 Taksonomi Proyeksi Geometri Bidang 99
Gambar 7. 4 Proyeksi Paralel 99
Gambar 7. 5 Bermacam-macam Proyeksi Orthographic..... 100
Gambar 7. 6 Proyeksi Isometric, Dimetric dan Trimetric 101
Gambar 7. 7 Proyeksi Oblique 102
Gambar 7. 8 Proyeksi Oblique Jenis Cavalier dan Cabinet..... 102
Gambar 7. 9 Proyeksi Perspektif 103
Gambar 7. 10 Mekanisme Proyeksi Perspektif 104
Gambar 8. 1 Ilustrasi Transformasi Sebuah Garis 107
Gambar 8. 2 Ilustrasi Rotasi 110
Gambar 9. 1 Ilustrasi Dlitasi..... 114
Gambar 9. 2 Ilustrasi Proses Shearing 116
Gambar 10. 1 Window 118
Gambar 10. 2 Pembagian dari baris segmen window Sutherland 122
Gambar 10. 3 Line Clipping 123
Gambar 10. 4 Polygon Clipping 124
Gambar 10. 5 Curve Clipping..... 125
Gambar 11. 1 Vektor dan Piksel 128

Gambar 11. 2 Line Thinning 129
Gambar 11. 3 Antialiasing..... 130
Gambar 11. 4 Interpolasi 131
Gambar 11. 5 Checkerboards..... 132
Gambar 11. 6 Editing Photosop Alias 133
Gambar 11. 7 Memperbesar Gambar 134
Gambar 11. 8 Video Gaming 135
Gambar 11. 9 Tipografi Piksel dan Alias 135
Gambar 11. 10 Suatu Logo Brand 136
Gambar 11. 11 Tipe Gambar Anti-aliased..... 136
Gambar 11. 12 Pencitraan berbasis piksel..... 137
Gambar 11. 13 Gambar menjadi media abstrak..... 137
Gambar 12. 1 System Koordinat 3D 140
Gambar 12. 2 CSG 143
Gambar 12. 3 CSG 3 Object..... 143
Gambar 12. 4 CGS Pengurangan..... 144
Gambar 12. 5 Lathe Object..... 144
Gambar 12. 6 Wireframe Rendering 146
Gambar 12. 7 Line Rendering..... 147
Gambar 12. 8 Shaded Rendering 147
Gambar 12. 9 Proses Rendering 148
Gambar 12. 10 Point of View 149
Gambar 13. 1 Program Pertama OpenGL..... 156
Gambar 13. 2 Diagram pemrosesan citradigital 159
Gambar 13. 3 Image enhancement..... 160
Gambar 13. 4 Image restoration 160
Gambar 13. 5 Color image processing..... 161

Gambar 13. 6 Wavelet dan multiresolution processing 161

Gambar 13. 7 Mage compression..... 161

Gambar 13. 8 Morphological processing..... 162

Gambar 13. 9 Segmentation..... 162

Gambar 13. 10 Object recognition 162

Gambar 14. 1 Output Animasi Orang..... 172

Gambar 14. 2 Output Aplikasi Komputer Grafik 174

PERTEMUAN 1

PENGENALAN GRAFIKA KOMPUTER

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu memahami pengenalan grafika komputer.

Pada pertemuan ini akan dijelaskan mengenai :

1. Pengatar Komputer Grafik
2. Gambaran Umum Mata Kuliah Grafika Komputer
3. Pengertian Pixel

B. URAIAN MATERI

1. Pengantar Komputer Grafik

a. Pengertian Komputer Grafik

Teknologi dan perkembangan ilmu saat ini sangat pesat. Dimana setiap liku-liku dan gerak-gerik kehidupan saat ini sangat dipengaruhi oleh adannya informasi yang kemudian informasi tersebut dapat diperoleh melalui ilmu teknologi dan pengetahuan.

Komputer adalah mesin yang dapat diinstruksikan untuk menjalankan urutan aritmatika atau operasi logika secara otomatis melalui pemrograman komputer, komputer digunakan sebagai alat pengoprasian atau penggunaan mesin untuk melakukan peloses perhitungan. Komputer dari generasi ke generasi mengalami perkembangan sangat pesat dalama mengolah data baik kecepatan maupun kapasitasnya. Sebuah Data yang dibuat tidak hanya angka atau teks, tapi juga komputer dapat mengolah sebuah gambar atau grafika. Komputer yang dapat mengaolah grafis untuk gambar atau grafika disebut grafis komputer.

Komputer grafik adalah ilmu dan seni berkomunikasi secara visual melalui tampilan komputer dan perangkat interaksinya. Aspek visual komunikasi biasanya dalam arah komputer-ke-manusia, dengan manusia-ke-

komputer arah yang dimediasi oleh perangkat seperti mouse, keyboard, joystick atau game pengontrol. Namun, ini pun mulai berubah. Data visual mulai mengalir kembali ke komputer, dengan antarmuka baru berdasarkan algoritme visi komputer yang diterapkan ke dalam video atau kamera. tetapi untuk arah komputer-ke-pengguna, konsumen utama komunikasi adalah manusia, dan dengan demikian cara manusia memandang citra sangat penting dalam desain garfish Grafik komputer adalah bidang lintas disiplin yang di dalamnya fisika, matematika, persepsi manusia, interaksi manusia-komputer, teknik, desain grafis, dan seni semuanya memainkan peran penting. Komputer grafik menggunakan fisika untuk model cahaya dan untuk melakukan simulasi untuk animasi. komputer grafik menggunakan matematika untuk mendeskripsikan bentuk. Komputer grafik menggunakan kemampuan persepsi manusia menentukan alokasi sumber daya. Komputer grafik menggunakan teknik dalam mengoptimalkan alokasi bandwidth, memori, dan waktu prosesor. Komputer grafik menggunakan Desain grafis dan seni berpadu dengan interaksi manusia-komputer untuk membuat arah komunikasi komputer-ke-manusia yang paling efektif.

Definisi sempit dari grafik komputer akan menyatakan bahwa itu mengacu pada pengambilan file model objek dalam sebuah adegan (deskripsi geometris dari hal-hal dalam sebuah adegan dan deskripsi tentang bagaimana mereka memantulkan cahaya) dan model cahaya yang dipancarkan ke scene (deskripsi matematis dari sumber energi cahaya, arah radiasi, distribusi panjang gelombang cahaya, dan lain-lain.), dan kemudian menghasilkan representasi pandangan tertentu dari pemandangan (cahaya yang sampai pada suatu mata imajine atau kamera di tempat kejadian). mengalikan cahaya yang masuk dengan reflektivitas objek di dalamnya untuk menghitung cahaya yang meninggalkan permukaan objek tersebut dan mengulangi prosesnya (memperlakukan permukaan sebagai sumber cahaya baru dan secara rekursif menjalankan operasi pengangkutan cahaya), menentukan semua cahaya yang pada akhirnya mencapai kamera. (Di prakteknya, pendekatan ini tidak bisa diterapkan, tetapi idenya tetap.) Sebaliknya, sistem komputer sama dengan suatu pemandangan, sistem komputer bertanggung jawab untuk menentukan iluminasi atau konten adegan (yang sistem grafis kemudian dapat “berkembang biak” bersama untuk mereproduksi gambar yang sama). Sebenarnya, tentu saja, sistem

penglihatan tidak dapat menyelesaikan masalah seperti yang dinyatakan dan biasanya bekerja dengan asumsi tentang pemandangan, atau pencahayaan, atau keduanya, dan mungkin juga memiliki beberapa tampilan pemandangan dari kamera yang berbeda, atau beberapa tampilan dari satu kamera tetapi pada waktu yang berbeda.

Faktanya, grafik jauh lebih kaya dari pada proses perkalian umum dalam meliat tampilan, seperti halnya penglihatan lebih kaya dari pada faktorisasi Sebagian besar penelitian saat ini dalam grafik adalah metode untuk membuat model geometris, metode untuk mewakili reflektansi permukaan

b. Sejarah Komputer Grafik

Sejarah komputer grafik telah ditemuka pada abad dahulu kala itu ketika bangsa Yunani, iRoma dan Mesir berkomunikasi secara grafik. Beberapa lukisan yang terdapat pada sebuah batu nisan. Orang-orang Mesir dapat dikatakan sebagai lukisan teknik. Perkembangan komputer grafik secara sederhana dapat dibagi menjadi empat fase, yaitu :

1) Fase Pertama (1950-1960)
   a) Tampilan grafis komputer Vectorscope pada komputer di MIT.
   b) Sistem Sage menggunakan pena cahaya pertama sebagai perangkat input di MIT Lab.
   c) Laboratorium Lawrence Livermore mengasosiasikan tampilan grafis dengan IBM 704 dan perekam film untuk gambar berwarna.
   d) Di National Bureau of Standards, foto yang diproses gambar pertama digunakan..
2) Fase Kedua (1960-1970)
   a) Di Laboratorium MIT Lincoln, menghasilkan sistem komunikasi grafis manusia-mesin.
   b) Komputer analog dikembangkan oleh pertamakali dan digunakan untuk mengubah gambar.
   c) IBM 2250, komputer grafis pertama yang tersedia.
   d) Game komputer grafik pertama, "Odyssey".
   e) Animasi komputer pertama "FLEXIPEDE" di Inggris.
3) Fase Ketiga (1970-1980)
   a) Konsep algoritma Z-buffer dan pemetaan tekstur dikembangkan.

b) Edwin Catmull memperkenalkan sistem animasi Tween.
c) Film penampilan 3D pertama dibuat oleh Gary Demos, John Whitey Jr. di Triple-I.
d) Komputer digital pertama digunakan dalam grafik komputer di Digital Equipment Corporation (DEC).

4) Fase Keempat (1980-1990)
a) Penggunaan luas pertama dari animasi grafik 3D dilakukan dalam film unggulan Disney.
b) AutoCAD 1.0 diluncurkan-Ini hanya digunakan untuk representasi bingkai kawat.
c) Perangkat lunak pencitraan medis yang dikombinasikan dengan teknologi Voxel.
d) Standar Video Graphics Array (VGA) diperkenalkan.

2. Gambaran Umum Mata Kuliah Komputer Grafik

Grafika komputer bukan ilmu yang berdiri sendiri. Grafik komputer adalah cabang ilmu komputer yang berhubungan dengan menghasilkan gambar dengan bantuan computer. Saat ini, grafik komputer adalah teknologi inti dalam fotografi digital, film, permainan video, telepon seluler dan layar komputer, dan banyak aplikasi khusus. Banyak perangkat keras dan perangkat lunak khusus telah dikembangkan, dengan tampilan sebagian besar perangkat digerakkan oleh perangkat keras grafis komputer. Pada dasarnya banyak ilmu yang menunjang sekaligus menjadi dasar grafika komputer, misalnya ilmu matematika, geometri, analisis atau metode numerik dan ilmu komputasi.

Table 1. 1 Kaitan Ilmu Grafika komputer dengan Ilmu Lain

Input atau Output	**Citra**	**Deskripsi**
Gambar	Pengolah Citra	Visi Komputer
Deskripsi	Grafika Komputer	AI

a. Peranan Dan Pengunaan Komputer Grafik

Grafika komputer telah menunjukkan kemajuan yang pesat dalam pengembangan berbagai aplikasi untuk menghasilkan gambar. Walaupun pada awalnya aplikasi dalam engineering dan sains memerlukan peralatan yang sangat mahal, perkembangan teknologi komputer memberikan kemudahan penggunaan komputer sebagai alat bantu aplikasi grafik komputer interaktif. Pada saat ini grafika komputer digunakan secara rutin dibidang ilmu pengetahuan, industri, kedokteran, bisnis, teknik, pemerintahan, seni, hiburan, pendidikan, periklanan, dan lain sebagainya.

b. Aplikasi – aplikasi Komputer Grafik

Setidaknya ada tujuh area dimana aplikasi grafika komputer dapat dimanfaatkan, yaitu :

1) Computer Art

Menggunakan grafik komputer kita dapat membuat seni rupa dan komersial yang mencakup fasilitas untuk mendesain bentuk objek dan menentukan gerakan objek. Gambar kartun, lukisan, desain logo juga bisa dilakukan.

2) Computer Aided Drawing

Perancangan bangunan, mobil, pesawat terbang dilakukan dengan bantuan Computer Aided Drawing, ini membantu dalam memberikan detail kecil pada gambar dan menghasilkan gambar yang lebih akurat dan tajam dengan spesifikasi yang lebih baik.

3) Presentation Graphics

Untuk penyusunan laporan atau rangkuman data keuangan, statistik, matematis, ilmiah, ekonomi untuk laporan penelitian, apalagi pembuatan grafik batang, diagram lingkaran, diagram waktu, dapat dilakukan dengan menggunakan alat bantu yang terdapat pada grafik komputer.

4) Entertainment

Grafik komputer menemukan sebagian besar kegunaannya dalam industri film dan industri igame. Digunakan untuk membuat film, video musik, acara televisi, film animasi kartun. Dalam industri game di mana fokus dan interaktivitas adalah pemain utamanya, grafik komputer membantu menyediakan fitur-fitur tersebut dengan cara yang efisien.

5) Education

Model yang dihasilkan komputer sangat berguna untuk mengajarkan sejumlah besar ikonsep dan dasar dengan cara yang mudah dipahami dan dipelajari. Dengan menggunakan grafik komputer, banyak model pendidikan dapat dibuat di mana lebih banyak minat dapat dihasilkan di antara siswa tentang subjek tersebut.

6) Training

Sistem khusus untuk pelatihan seperti simulator dapat digunakan untuk melatih kandidat idengan cara yang dapat dipahami dalam waktu singkat dengan pemahaman yang lebih baik. Pembuatan modul pelatihan menggunakan grafik komputer sederhana dan sangat bermanfaat.

7) Visualisation

Saat ini kebutuhan akan visualisasi telah meningkat secara drastis, kebutuhan visualisasi dapat dilihat pada banyak teknologi canggih, visualisasi data membantu idalam menemukan wawasan data, untuk memeriksa dan mempelajari perilaku proses di sekitar kita diperlukan visualisasi yang sesuai yang dapat dicapai. melalui penggunaan grafik komputer yang tepat

8) Image Processing

Berbagai jenis foto atau gambar memerlukan pengeditan agar dapat digunakan di tempat yang berbeda. Pemrosesan gambar yang ada menjadi yang lebih halus untuk interpretasi yang lebih baik adalah salah satu dari banyak aplikasi grafik komputer.

9) Machine Drawing

Grafik komputer sangat sering digunakan untuk merancang, memodifikasi dan membuat berbagai bagian mesin dan seluruh mesin itu sendiri, alasan utama di balik ipenggunaan grafik komputer untuk tujuan ini adalah ketepatan dan kejelasan yang kami dapatkan dari gambar seperti itu adalah yang utama dan sangat diinginkan untuk pembuatan mesin yang aman menggunakan gambar-gambar ini.

10) Graphical User Interface

Penggunaan gambar, gambar, ikon, menu pop-up, objek grafis membantu dalam imenciptakan lingkungan yang ramah pengguna di mana bekerja menjadi mudah dan menyenangkan, menggunakan grafik komputer kita dapat menciptakan suasana di mana semuanya

bisa otomatis dan siapa saja bisa mendapatkan yang diinginkan tindakan dilakukan dengan cara yang mudah

c. Perkakas Grafik Komputer

Sebagai bidang ilmu yang tidak hanya berdasarkan teori namun juga memiliki ipenerapan yang tinggi. grafika komputer memerukan sumber daya manusia, perakat lunak dan perakat kelas. Kebutuhan minimal untuk perangkat keras antara lain yaitu Komputer, Display, Printer, Monitor, Webcam, dan sebagainya. Yang di perlukan Perangkat lunak dalam mengembangkan sebuah komputer grafik antara lain yaitu:

1) Sistem operasi Perangkat lunak yang menunjang antara lain yaitu Windows 7 atau 8, Windows XP, Vista
2) Aplikasi Perangkat lunak pembuatan gambar dan edit gambar yang menunjang antara lain yaitu Adobe Illustrator, Photoshop dan sebagainya
3) Animasi dan pembelajaran grafika Perangkat lunak, yang menunjang antara lain yaitu FlipaClip , 3D Studio Max
4) Perangkat lunak visualisasi, yang menunjang antara lain yaitu Matlab dan Maple
5) Perangkat lunak untuk virtual reality yang menunjang antara lain yaitu augmented reality dan sebagainya
6) Perangkat Lunak pemrograman yang menunjang antara lain yaitu android studio.

Dalam komputer grafik beberapa istilah penting yang diketahui adalah antara lain vector graphics dan pixel, LED, LCD, CRT, VGA, CGA, EGA, SVGA, XGA, JPG, PCX, GIF, Bitmap, Plasma, BMP, Resolition, Animation, Modeling, wireframe, Rendering,

d. Teknologi Output

Pada dasarnya perangkat keras dan perangkat lunak telah tersedia untuk kelengkapan berbagai aplikasi grafik. Kemampuan untuk menyajikan bentuk 2D dan 3D merupakan hal yang sudah biasa dalam aplikasi grafik pada dasarnya. Dengan Personal Computer (PC), dapat digunakan untuk berbagai macam alat input interaktif.

1) Teknologi Display

Teknologi Display yang dioperasikan secara elektrik telah berkembang dari sistem elektromekanis untuk menampilkan teks, hingga perangkat elektronik yang mampu menampilkan grafik warna 3D gerak penuh. Perangkat elektromagnetik, menggunakan kumparan solenoid untuk mengontrol flag atau flap yang terlihat, adalah tipe yang paling awal, dan digunakan untuk tampilan teks seperti harga pasar saham dan waktu tampilan kedatangan atau keberangkatan. Cathode ray tube (CTR) adalah perangkat keras dari teknologi tampilan teks dan video selama beberapa dekade hingga digantikan oleh plasma, kristal cair (LCD), dan perangkat solid-state seperti transistor film tipis (TFT), LED dan OLED. Dengan munculnya transistor efek medan logam-oksida-semikonduktor (MOSFET), chip sirkuit terintegrasi (IC), mikroprosesor, dan perangkat mikroelektronik, lebih banyak lagi elemen gambar individual ("piksel") dapat digabungkan ke dalam satu perangkat tampilan, memungkinkan grafik layar dan video.

2) Random-scan Display

Dalam Random-Scan Display berkas elektron diarahkan hanya ke bidang layar di mana gambar harus dibuat. Ini juga disebut tampilan vektor, karena menggambar satu baris pada satu waktu. Itu dapat menggambar dan menyegarkan baris komponen gambar dalam urutan tertentu. Pen plotter adalah contoh tampilan pemindaian acak.

Jumlah baris mengatur kecepatan refresh pada tampilan pemindaian acak. Area memori yang disebut file tampilan refresh menyimpan definisi gambar sebagai sekumpulan perintah menggambar garis. Sistem kembali ke perintah baris pertama dalam daftar, setelah semua perintah gambar telah diproses. Sistem vektor berkualitas tinggi dapat menangani sekitar 100, 00 garis pendek pada kecepatan penyegaran ini. Penyegaran yang lebih cepat dapat membakar fosfor. Untuk menghindari hal ini, setiap siklus penyegaran ditunda untuk mencegah kecepatan penyegaran lebih besar dari 60 bingkai per detik.

3) Monitor Color CRT

Dalam metode ini, layar CRT dilapisi dengan dua lapisan fosfor, merah dan hijau dan warna yang ditampilkan tergantung sejauh mana

berkas elektron menembus lapisan fosfor. Dalam metode ini, layar CRT dilapisi dengan dua lapisan fosfor, merah. dan hijau dan warna yang ditampilkan tergantung pada seberapa jauh berkas elektron menembus lapisan fosfor.

Monitor CRT ditampilkan dengan menggunakan kombinasi fosfor. Fosfor memiliki warna yang berbeda. Ada dua pendekatan populer untuk menghasilkan tampilan berwarna dengan CRT

a) Beam Penetration Method

Beam Penetration Method telah digunakan dengan monitor pemindaian acak. Dalam metode ini, layar CRT dilapisi dengan dua lapisan fosfor, merah dan hijau dan warna yang ditampilkan tergantung pada seberapa jauh elektron menembus lapisan fosfor. Cara ini menghasilkan empat warna saja, merah, hijau, oranye dan kuning. Elektron lambat hanya mengarahkan lapisan merah, maka layar hanya menunjukkan warna merah. elektron berkecepatan tinggi mengarahkan lapisan hijau bagian dalam. Dengan demikian layar menunjukkan warna hijau.

b) Shadow Mask Method

Shadow Mask Method digunakan dalam Raster-Scan System karena menghasilkan rentang warna yang jauh lebih luas daripada metode penetrasi sinar. Ini digunakan di sebagian besar perangkat TV dan monitor berwarna. Shadow-Mask Method memiliki 3 titik warna fosfor di setiap posisi piksel.

(1) Satu titik fosfor memancarkan: lampu merah
(2) Cahaya lain: lampu hijau
(3) Ketiga memancarkan: cahaya biru

CRT jenis ini memiliki 3 elektron, satu untuk setiap titik warna dan Shadow-Mask Method tepat di belakang layar berlapis fosfor.

3. Pengertian Pixel

Pixel adalah unit terkecil dari gambar atau grafik digital yang dapat ditampilkan dan direpresentasikan pada perangkat tampilan digital. Pixel adalah unit logika dasar dalam grafik digital. Piksel digabungkan untuk membentuk gambar, video, teks, atau hal yang terlihat lengkap di layar computer. Piksel juga dikenal sebagai elemen gambar (pix = picture, el = element).

Pixel diwakili oleh titik atau persegi pada layar tampilan monitor komputer. Piksel adalah blok bangunan dasar dari gambar atau tampilan digital dan dibuat menggunakan koordinat geometris. pixel tergantung pada grafik, monitor tampilan, kuantitas, ukuran dan kombinasi warna. piksel bervariasi dan diukur dalam kaitannya dengan resolusi layar. Misalnya, komputer dengan resolusi tampilan 1280 x 768 akan menghasilkan maksimal 98.3040 piksel pada satu layar tampilan.

Penyebaran resolusi piksel juga menentukan kualitas tampilan lebih banyak piksel per inci layar monitor menghasilkan hasil gambar yang lebih baik. Misalnya, gambar 2,1 megapiksel berisi 2.073.600 piksel karena memiliki resolusi 1920 x 1080.

Ukuran fisik suatu piksel berbeda-beda, bergantung pada resolusi layar. Ini akan sama dengan ukuran dot pitch jika layar disetel ke resolusi maksimumnya, dan akan lebih besar jika resolusinya lebih rendah karena setiap pixel akan menggunakan lebih banyak titik. Karena itu, setiap piksel dapat terlihat, mengarah ke gambar yang tebal dan tebal yang didefinisikan sebagai "piksel". Piksel disusun secara seragam dalam 2D, meskipun tersedia beberapa pola pengambilan sampel yang berbeda. Misalnya, di layar LCD, tiga warna utama diambil sampelnya di lokasi yang berbeda dari 2D, sedangkan kamera warna digital menggunakan 2D yang lebih teratur

Setiap pixel memiliki alamat logis yang unik, berukuran delapan bit atau lebih dan, di sebagian besar perangkat tampilan kelas atas, kemampuan untuk memproyeksikan jutaan warna berbeda. Warna setiap piksel ditentukan oleh pencampuran spesifik dari tiga komponen warna RGB. pada sistem warna, jumlah byte yang berbeda dapat dialokasikan untuk menentukan setiap komponen warna piksel. Misalnya, dalam sistem warna 8-bit, hanya satu byte yang dialokasikan per pixel, membatasi palet menjadi 256 warna saja. Dalam sistem warna 24-bit yang umum digunakan untuk hampir semua monitor PC dan

tampilan ponsel cerdas, tiga byte dialokasikan, satu untuk setiap warna skala RGB, yang menghasilkan total 16.777.216 variasi warna. Sistem warna dalam 30-bit mengalokasikan masing-masing 10 bit merah, hijau, dan biru untuk total 1,073 miliar variasi warna.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 1	1. Jelaskan pengertian grafik komputer menurut anda! 2. Tentukan kepanjangan dan atau definisi istilah-istilah grafis sebagai berikut a. CGA b. CRT c. LCD d. JPG e. GIF f. BMP 3. Sebutkan dan jelaskan perbedaan dari Beam Penetration Method dan Shadow-Mask Method!

D. REFERENSI

PERTEMUAN 2

TITIK DAN PALET WARNA

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan titik dan palet warna.

Pada pertemuan ini akan dijelaskan mengenai:

1. Membaca Dan Menulis Pixel
2. Membuat Palet Warna
3. RGB Dan HSV Perihal Himpunan

B. URAIAN MATERI

1. Membaca Dan Menulis Pixel

Piksel adalah satu titik data dalam gambar digital atau di monitor, dan pengukuran PPI menunjukkan jumlah piksel yang terkandung dalam gambar atau layar. Piksel per inci dinyatakan dengan menunjukkan jumlah piksel yang tersedia secara horizontal dengan jumlah yang tersedia secara vertikal. Dengan demikian, gambar dengan lebar 200 piksel dan lebar 200 piksel akan diekspresikan sebagai gambar 200 x 200 PPI. Meskipun ada pengecualian, sebagian besar perangkat mengandalkan piksel persegi dalam mengambil dan menampilkan gambar. Pada ujung tertinggi pada skala resolusi, mesin cetak gambar berwarna akan dapat menghasilkan hasil cetak yang memiliki lebih dari 2.500 titik per inci dengan pilihan 16 juta warna lebih untuk setiap inci, dalam istilah komputer berarti gambar seluas per satu inci persegi yang bisa ditampilkan pada tingkat resolusi tersebut sepadan dengan 150 juta bit informasi. Monitor atau Layar yang sering kita temui terdiri dari ribuan piksel yang terbagi dalam sebuah baris-baris dan kolom-kolom. Jumlah piksel yang terdapat pada sebuah monitor dapat kita ketahui dari resolusinya. Resolusi maksimum yang disediakan oleh monitor adalah 1024x768, maka jumlah pixel yang ada di sebuah layar monitor tersebut adalah 786432 piksel. Semakin tinggi jumlah

piksel yang tersedia maka dalam monitor, semakin tajam gambar yang mampu ditampilkan oleh monitor tersebut.

Resolusi adalah kehalusan detail dalam gambar bitmap dan diukur dalam piksel per inci (ppi). Semakin banyak piksel per inci, semakin besar resolusinya. Umumnya, gambar dengan resolusi lebih tinggi menghasilkan kualitas gambar cetakan yang lebih baik jumlah data gambar yang sama. menentukan setiap piksel. pada skala resolusi Sebuah piksel mengandung 8 bit (1 byte) jika dalam BW (hitam dan putih). Untuk gambar berwarna menggunakan skema warna tertentu yang disebut RGB (Merah, Hijau, Biru) yang direpresentasikan sebagai masing-masing 1 byte atau 24 bit (3 byte) per piksel. Ini juga disebut sebagai kedalaman bit gambar. Untuk menentukan kedalaman bit, Anda memerlukan jumlah bit yang digunakan untuk menentukan setiap pixel.

Piksel per inci (PPI) adalah ukuran resolusi dalam gambar digital atau tampilan video. Piksel adalah area iluminasi atau warna pada layar atau gambar komputer. PPI mengukur resolusi tampilan, atau kepadatan piksel, dari monitor atau layar komputer. Ukuran tersebut juga digunakan untuk menunjukkan resolusi gambar digital, serta kapasitas resolusi kamera atau pemindai yang menangkap gambar.

PPI mengukur jumlah piksel per baris per inci dalam foto digital. Angka ini secara langsung berkaitan dengan jumlah megapiksel yang dapat ditangkap kamera digital. Misalnya, Canon Digital Rebel asli adalah kamera 6,3 megapiksel dan menangkap 2048 vertikal dengan 3072 piksel horizontal. Karenanya, saat mencetak gambar 4x6, PPI akan menjadi 3072 piksel. 6 inci = 512 PPI. Itu cukup tinggi untuk mencetak foto 4x6 yang sangat detail. Namun demikian, jika Anda mencetak gambar poster 20x30 besar dari gambar 6,3 megapiksel, PPI akan menjadi 3072 px. / 30 inci = 102,4 PPI.

2. Membuat Palet Warna

a. Pengertian Palet Warna

Istilah lain dari Palet Warna yaitu colour palet warna dimana sebagai perpaduan warna. palet warna adalah suatu himpunan warna akan disusun dan dicampur hingga menjadikan kombinasi warna yang sangat indah.

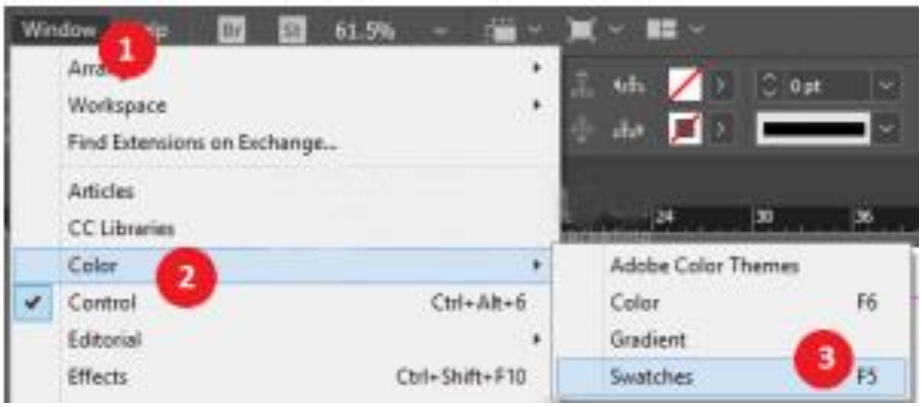
Terdapat banyak dasar suatu gambar bisa disebut indah dan cantik

untuk dipandang. *satu satu* di antara yang ada adalah warnanya yang indah. Warna yang indah tersebut bisa tercipta karena terdapat beberapa warna yang dikombinasikan menjadi suatu gambar. Dengan kata lain warna yang indah terjadi karena adanya kombinasi warna yang tepat**.** Kombinasi warna tersebut bisa terdiri dari 2 hingga 5 warna yang serasi. Pada ilmu desain Kombinasi warna ini biasa juga disebut dengan **palet warna** atau ***Color Palette***

Warna adalah karakteristik objek yang didapatkan dari hasil pantulan sinar tampak dari objek tersebut. Pada citra digital, warna pada suatu pixel direpresentasikan dengan tiga warna, yaitu warna merah, hijau, dan biru. Besaran nilai ketiga warna tersebut bergantung banyaknya bit yang diperlukan untuk merepresentasikan warna tersebut. Semakin banyak bit yang digunakan, semakin banyak rentang nilai merah, hijau, dan biru yang dapat digunakan untuk merepresentasikan warna. Terdapat dua tipe warna yang umum yang dapat digunakan, yaitu tipe warna RGB dan tipel warna CMY. Pada model RGB, warna direpresentasikan dengan kombinasi nilai warna merah, hijau, dan biru. Sedangkan Model CMYK merepresentasikan warna dengan nilai magenta, cyan, dan yellow. Cyan, magenta dan yellow adalah warna komplementer dari warna merah, hijau, dan biru. Kedua model ini digunakan pada hardware komputer. Untuk mendapatkan warna komplementer, dilakukan kombinasi warna pokok dengan proporsi tertentu untuk mendapatkan warna lain

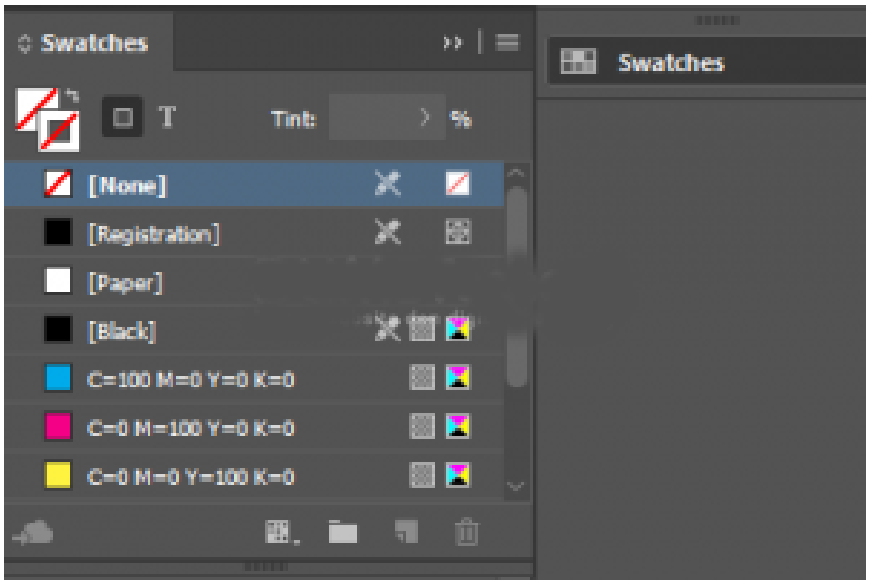
b. Cara Membuat Palet Warna

1) Buka Adobe InDesign, dan buka File > New > Document, dan buat dokumen dengan ukuran apa pun.
2) Lalu untuk menampilkan panel swatches klik menu Window > Color > Swatches dan pilih Swatch Warna Baru dari menu drop-down panel (dapat diakses di sudut kanan bawah).



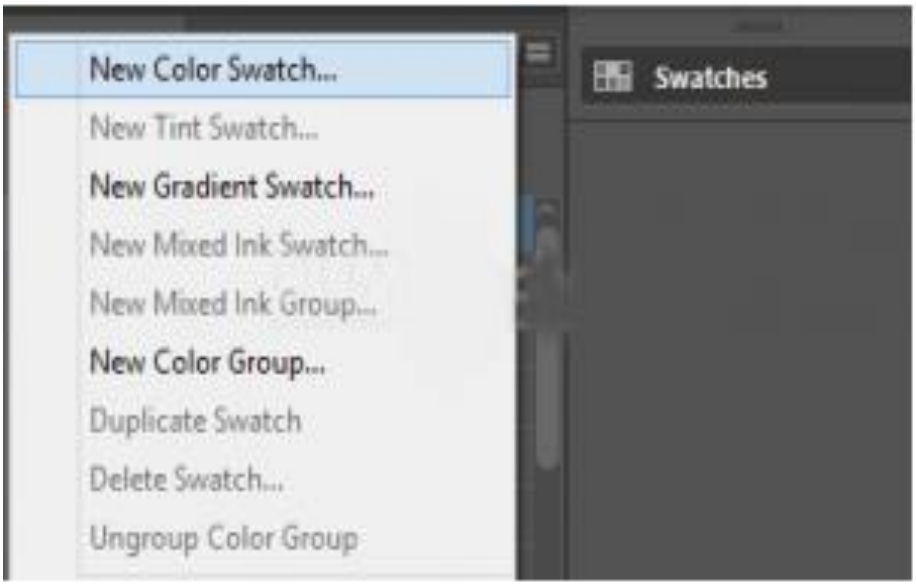
Gambar 2. 1 Menampilkan panel swatches

Lalu akan muncul panel seperti gambar dibawah ini



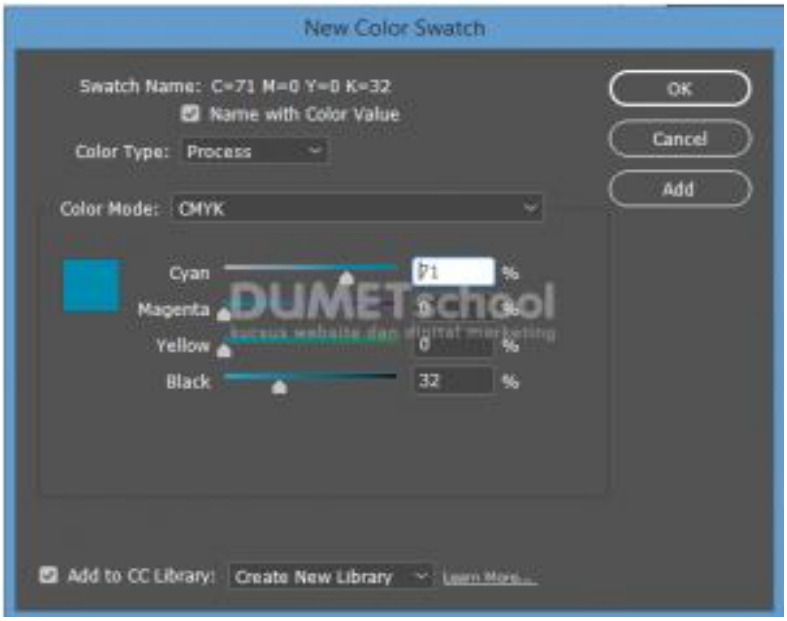
Gambar 2. 2 Menu swatches

3) Untuk menambahkan warna baru pada swatches. Klik ikon yang berada pada pojok kanan atas pada panel, lalu pilih new color swatch.



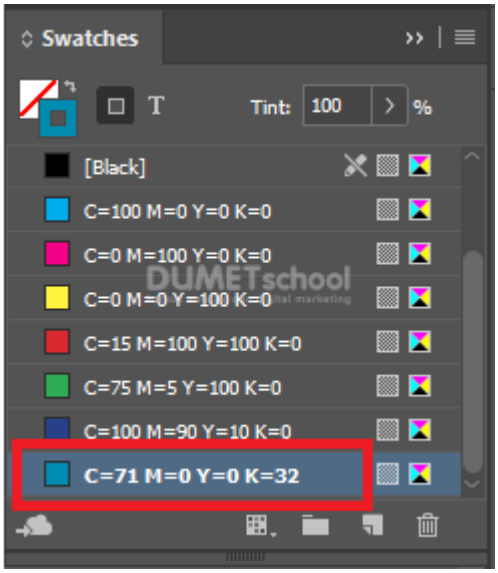
Gambar 2. 3 Menambahkan warna baru pada swatches

Lalu akan muncul pengaturan seperti gambar di bawah ini, teman-teman bisa mengatur CMYK untuk menentukan warna apa yang akan dibuat, dan jika telah selesai lalu klik OK.



Gambar 2. 4 Menentukan warna baru pada swatches

4) Dan warna yang telah dibuat tadi berada pada paling bawah panel.



Gambar 2. 5 Warna yang telah dibuat

3. RGB Dan HSV

a. RGB (Red Green Blue)

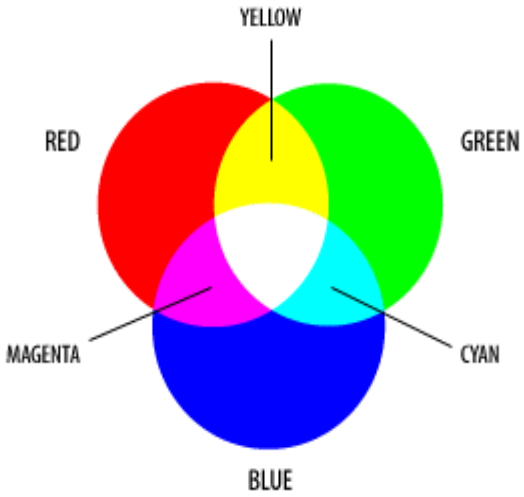
Model warna RGB adalah model warna aditif di mana warna merah, hijau dan biru dicampur bersama dalam berbagai proporsi untuk membentuk susunan warna yang berbeda. Nama itu diberikan dengan huruf pertama dari tiga warna primer merah, hijau dan biru. Dalam model ini, warna disiapkan dengan menambahkan komponen, dengan putih memiliki semua warna di dalamnya dan hitam tanpa kehadiran warna apa pun. Model warna RGB digunakan dalam berbagai tampilan digital seperti TV dan tampilan video, tampilan komputer, kamera digital dan jenis perangkat tampilan berbasis cahaya lainnya.

Model warna adalah proses untuk menciptakan lebih banyak warna menggunakan beberapa warna primer. Ada dua jenis model warna yang digunakan, yaitu model warna Additive dan model warna subtraktif. Pada warna aditif, lampu model digunakan untuk menampilkan warna. Sedangkan pada model warna subtraktif, tinta cetak digunakan untuk menghasilkan warna. Model warna aditif yang paling umum digunakan adalah model warna RGB dan model warna CMYK digunakan untuk pencetakan.

Model warna RGB adalah model warna aditif yang menggunakan warna Merah, Hijau dan Biru. Penggunaan utama model warna RGB adalah

untuk menampilkan gambar di perangkat elektronik. Pada proses model warna RGB ini jika ketiga warna tersebut ditumpangkan dengan intensitas paling kecil maka terbentuk warna hitam dan jika ditambah dengan intensitas cahaya penuh maka terbentuk warna putih. Untuk membuat susunan warna yang berbeda, warna primer ini harus ditumpangkan dalam intensitas yang berbeda. Menurut beberapa penelitian, intensitas setiap warna primer dapat bervariasi dari 0 hingga 255 dan menghasilkan hampir 16.777.216 warna.

Prinsip dasar di balik kerja model warna RGB adalah pencampuran warna aditif. Ini adalah proses pencampuran 3 warna primer merah, hijau dan biru bersama-sama dalam proporsi yang berbeda untuk menghasilkan warna yang lebih berbeda. Untuk setiap warna primer, dimungkinkan untuk mengambil 256 corak berbeda dari warna itu. Jadi dengan menambahkan 256 shade dari 3 warna primer, kami dapat menghasilkan lebih dari 16 juta warna berbeda. Sel kerucut atau fotoreseptor adalah bagian dari mata manusia yang bertanggung jawab untuk persepsi warna. Dalam model warna RGB, karena kombinasi warna primer menciptakan warna berbeda yang kita rasakan dengan merangsang sel kerucut yang berbeda secara bersamaan.



Gambar 2. 6 Model warna RGB

b. Kegunaan model warna RGB

Mode warna RGB digunakan untuk warna layar atau warna pendar berupa pancaran cahaya. Sebetulnya, warna ini hampir mirip dengan teori warna dasar (merah, kuning, biru), namun warna hijau sebagai pengganti

kuning di sini. Kegunaan utama dari warna RGB sebagai warna pokok adalah untuk menampilkan gambar pada media elektronik. Namun sebelum era teknologi pun, mode warna ini telah mendapat landasan khusus karena pemahaman manusia terhadap teori trikomatik.

1) Contoh Penerapan Mode Warna RGB dalam Berbagai Tampilan

Beberapa aplikasi mode warna RGB adalah sebagai berikut :

a) CRT

CRT yang berwarna menggunakan layar phospor dengan mode RGB, perbandingannya adalah sebanyak 33% untuk warna merah, 11% untuk hijau, dan 56% untuk biru.

b) LCD

Pada tampilan layar LCD, mode warna RGB akan dipadukan dengan IC untuk menampilkan dan membedakan warna dengan spektrum yang sesuai dengan mode warna tersebut.

c) Layar Plasma

Layar plasma seperti monitor komputer dan televisi menggabungkan 3 warna pokok RGB menjadi 1 piksel atau titik hingga menghasilkan gambar pada layarnya.

2) Mengkonversi warna RGB ke HSV

Rumus konversi RGB ke HSV:

Nilai R , G , B dibagi 255 untuk mengubah rentang dari 0..255 hingga 0..1:

$R' = R / 255$

$G' = G / 255$

$B' = B / 255$

$Cmaks = \text{maks}(R', G', B')$

$Cmin = \text{min}(R', G', B')$

$\Delta = Cmaks - Cmin$

menghitungan Hue :

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

menghitungan Saturation::

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

menghitungan Value:

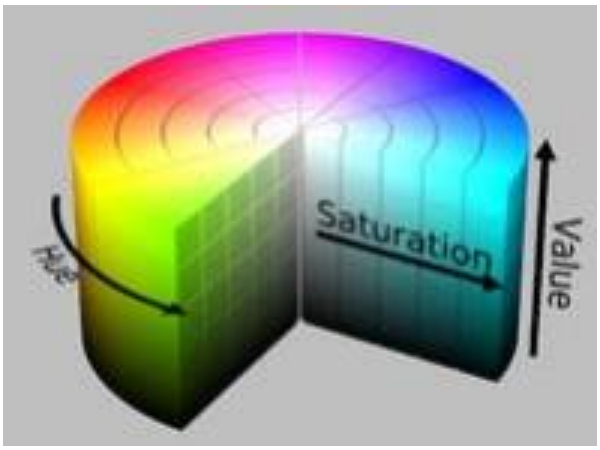
V = Cmaks

c. HSV

Dalam model warna HSV, warna ditentukan oleh hue(H), saturation (S) dan value (V), lebih menyerupai persepsi warna manusia dari pada model warna aditif dan subtraktif . Mudah untuk menyesuaikan warna dengan terang dan kecerahannya. Karenanya, dalam sebagian besar aplikasi grafik, Anda dapat menentukan warna sesuai dengan model warna HSV.

hue (H) diberikan sebagai sudut pada lingkaran berwarna, oleh karena itu dapat mencapai nilai antara 0 ° dan 360 °. 0 ° untuk warna merah, 120 ° untuk warna hijau dan 240 ° untuk warna biru. The saturasi (S) dinyatakan sebagai persentase dan karena itu dapat mencapai nilai antara 0% dan 100% (atau 0-1). Saturasi 100% berarti warna yang benar-benar jenuh dan murni, semakin kecil terangnya, semakin warna berubah menjadi abu-abu netral. Nilai terang atau gelap (V) juga diberikan sebagai persentase, di mana 0% berarti tidak ada kecerahan (karenanya hitam) dan kecerahan penuh 100%, maka spektrum antara warna murni (saturasi 100%) dan putih (saturasi 0 %). Jika keduanya, saturasi dan kecerahannya

100%, hasil warna murni . Jika saturasi 0% dan kecerahan 100% berarti putih dan untuk semua kasus di mana kecerahan 0% berarti hitam.



Gambar 2. 7 Warna HSV

d. Mengkonversi warna HSV ke RGB

Rumus konversi RGB ke HSV:

$$\left[\frac{h}{60}\right] \quad mod6$$

$$f = \frac{h}{60} - h_i$$

$$P = v\,x(1 - s)$$

$$q = v\,x\,(1 - f\,x\,s)$$

$$t = v\,x\,(1 - (1 - f)x\,s$$

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 2	1. Jelaskan perbedaan antara model warna RGB dan HSV? 2. Jelaskan Kenapa RGB merupakan warna cahaya

D. REFERENSI

PERTEMUAN 3

ALGORITMA GARIS DAN POLIGON

A. TUJUAN PEMAHAMAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan algoritma garis dan poligon.

Pada pertemuan ini akan dijelaskan mengenai:

1. Algoritma Pembentukan Garis
2. Algoritma Garis Bresenham
3. Membuat Poligon

B. URAIAN MATERI

1. Algoritma Pembentukan Garis

Ilmu grafis merupakan salah satu bentuk dari geometri paling mendasar yang umum digunakan ketika membuat seuuah bentuk/objek yang memiliki struktuur kompleks. Titik dan juga garis lurus merupakan salah satu contoh bentuk geometri dari sebuah gambar. Kedua hal ini yaitu titik dan juga garis merupakan contoh output dalam pembentukan sebuah gambar, misalnya persegi, lingkaran, kerucut, kurva dan juga gambar-gambar permukaan yang memiliki bentuk lengkung, karakter, area warna dan hal-hal lainnya.

Keberadaan titik dan garis ddiyakini sudah ada sejak masa lampau/zaman nenek moyang, hal ini diiduukung dari kebudayaan mereka ketika mengadaka sebuah upacara/perayaan dalam sistem kepercayaan mereka yaitu dengan cara membuat goresan di dalam dinding gua. Berawal dari goresan inilah kemudian dihasilkan berbagai macam garis yang saling tersambung dan nantinya akan membentuk sebuah obyek yang biasa disebut dengan gambar. Kemudian, dari gambar ini terciptalah lukisan yang akan menjadi awal mula dari kemunculan beragam jenis huruf/alfabet yang digunakan sebagai media komunikasi oleh manusia. Selain bentuk garis yang dikenal dengan kontur, garis juga umum

digunakan untuk menjelaskan sebuah gerak maupun bentuk, dalam berbagai bentuk antara lain, bentuk dua dimensi dan tiga dimensi.

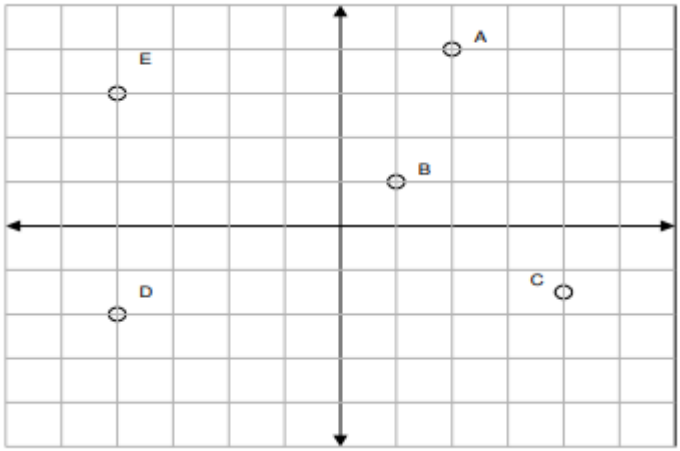
Untuk mengenal dan mengetahui ilmu tentang Algoritma Pembentukan Garis, maka kita mulai dari latar belakang dan juga sejarah dari algoritma itu sendiri. Algoritma merupakan inti utama dari ilmu informatika. Terdapat beragam ilmu pengetahuan khususnya ilmu perkomputeran mengambil dasar/mengacu ke dalam terminologi algoritma, salah satunya adalah algoritma routing di dalam sebuah jaringan komputer, kemudian ada algoritma bresenham yang digunakan dalam membuat garis lurus di dalam grafika komputer, selanjutnya terdapat algoritma Knuth-Morris-Pratt yang mempunyai kegunaan untuk mencari sebuah pola dalam sebuah teks (information retrievel), dan contoh-contoh lainnya.

Jika ditelaah dari awal mula katanya, kalimat “algoritma” memiliki sejarah tersendiri. Kita akan mendapatkan kosakata *algorsim* atau kurang lebih memiliki pengertian sebagai kegiatan perhitungan menggunakan angka – angka. Cukup banyak ilmuwan yang berusaha keras mencari darimana awal mula kosakata ini (algorism) berasal, akan tetapi hasil yang didapatkan tidaklah memuaskan. Kata algorism berasal dari nama seorang ilmuwan dari jazirah arab yang sangat terkenal akan karya-karyanya, yaitu Al-Khuwarizmi. Ia membuat sebuah buku yang diberi judul Kitab “al jabar wal-muqabala”, buku ini jika diartikan kuarng lebih memilikii arti “Buku Pemugaran dan Pengurangan”. Awal mula kata ‘aljabar’ (*algebra*) juga didapatkann dari buku tersebut.

a. Titik

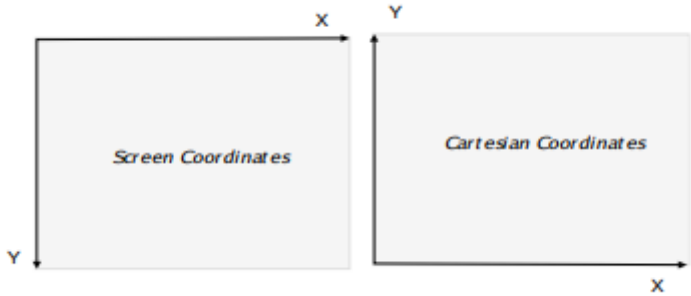
Sebuah titik dalam cabang ilmu grafika komputer dapat didefinisikan sebagai letak/posisi objek didalam sebuah *Coordinates System*. Pola titik kordinat yang pada umumnya digunakan adalah kordinat polar ataupun kordinat kartesius. Ketika kita menggunakan ilmu grafika komputer, cara paling sering yaitu adalah dengan menggunakan kordinat kartesius. Dalam penggunaannya, posisi titik difungsikan untuk mengetahui letak/posisi dari gabungan dari angka-angka untuk selanjutnya menetapkan letak titiknya dengan menggunakan kordinat posisi x dan juga y.

Sebagai contohnya adalah, Ketika kita meletakan posisi titik sebagai berikut : titik A berada di kordinat (2,4), titik B (1,1), titik C (4,-1.5), titik D (-4,-2), dan titik E (–4,3) dalam koordinat cartesius, maka kita dapat menggambarkannya seperti dibawah ini:



Gambar 3. 1 Letak titik dalam koordinat cartesius

Terdapat 2 (dua) koordinat didalam komputer khusunya yang umum digunakan oleh Operation Systems (OS) dalam Windows, yang pertama adalah kordinat layar (*Screen Cordinates*), kedua adalah kordinat kartesius (*Cartesian Cordinates*), perbedaan diantara dua sistem kordinat ini seringkali membuat kita bingung. Maka dari itu kita bisa mencoba memperhatikan contoh dibawah ini :

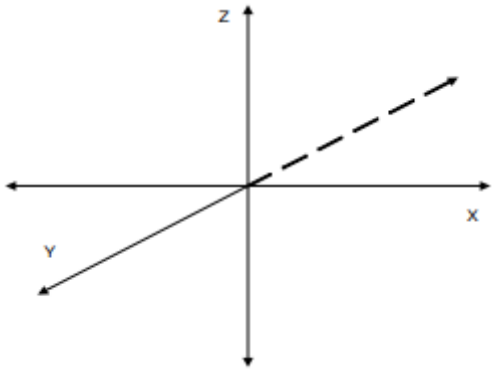


Gambar 3. 2 Perbandingan antara Cartesian dan Screen Cordinates

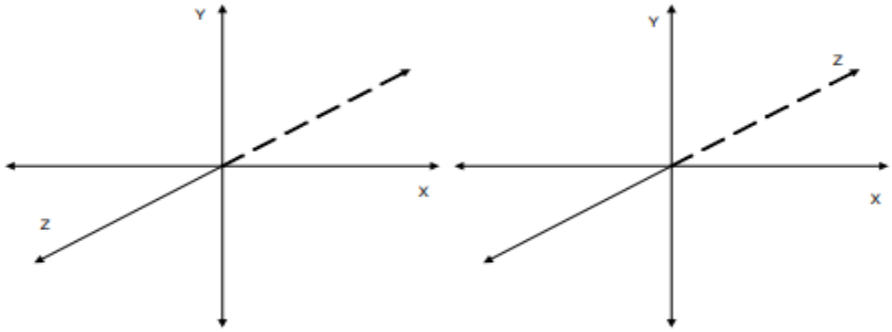
Secara umum prinsip monitor memang didesain untuk membuat garis dari bagian atas menuju bagian bawah, maka dari itu letak titik y pada kordinat kartesius dan kordinat layar memiliki titik sumbu yang berbeda, dalam kordinat kartesius posisi sumbu titik Y mengarah ke atas, sebaliknya *untuk kordinat* layar, posisi sumbu titik Y memiliki arah ke bawahh. Pada umumnya ketika kita akan menggunakan teknik *rendering pipeline*, hal

utama yang perlu kita lakukan yaitu mengubah posisi *Cartesian Coordinates* menjadi *Screen Coordinates*.

Untuk Operation Systems (OS) Linux, posisi koordinat Y yang dipakai baik *Cartesian* maupun *Screen Coordinates*, memiliki kesamaan yaitu Y memiliki arah positif kearah atas. Sedangkan dalam koordinat 3D, tidak berbeda seperti dalam 2D, akan tetapi terdapat tambahan 1 (satu) titik lagi yang menjadi sumbu yaitu titik z atau biasa disebut sebagai axis-z. Setidaknya terdapat 2 teknik dalam membuat gambar sumbu di titik X, Y kemudian Z. Seperti kita lihat pada gambar, yaitu sumbu titik z memiliki arah menuju atas (gambar 3.3). Kedua adalah dengan koordinat sumbu y mengarah ke atas (gambar 3.4).



Gambar 3. 3 Posisi koordinat



Z mengarah keatas

Gambar 3. 4 Posisi koordinat Y mengarah keatas

b. Garis

Hal pertama yang akan kita pelajari pada pembahasan ini adalah apa itu garis ?. Garis adalah sebuah kumpulan titik, Contohnya adalah saat kita akan membuat sebuah garis lurus, maka hal utama yang perlu diketahui adalah posisi titik awal dan titik akhir dari garis tersebut. Setelah kita mengetahuinya, maka kita bisa menyambungkan titik awal dengan akhir sehingga membentuk sebuah garis lurus.

Persamaan garis lurus umumya dikenal dan dinotasikan dengan persamaan y = mx + c. Simbol *m* disini merupakan gradien yang merupakan proses dari pembagian antara titik delta Y dan juga delta X. Sedangkan simbol *c* merupakan konstanta. Hal ini menjadi fondasi dasar dalam algoritma yang digunakan dalam pembuatan garis lurus. Menggunakan rumus ini juga kita akan memulai mempelajari algortima yang secara umum dipakai untuk pembuatan sebuah garis lurus.

Saat ini, performa komputasi prosesor komputer telah sangat cepat. Sehingga hal ini juga berpengaruh terhadap kemampuan komputasi yang semakin handal. Oleh karena itu saat ini semakin banyak program aplikasi yang memakai grafik 3 dimensi yang cukup kompleks. Keadaan seperti ini tentunya perlu diimbangi dengan efisiensi algoritma yang ada, jadi pembuatan garis dan juga kurva dalam pembuatan grafik dapat menghasilkan gambar dengan kualitas maksimal.

Dalam menggambarkan sebuah pembuatan garis, baik dari Langkah awal sampai kepada Langkah akhir terdapat bermacam algorima yang dapat digunakan. Beberapa algoritma yang umum digunakan antara lain adalah algoritma Brute Force, DDA (Digital Differential Analyzer) dan Bressenham.

c. Algoritma Brute Force

Penggunaan algoritma ini dalam pembuatan sebuah gambar garis memiliki dasar dengan kedudukan titik (2-6), berikut penjelasannya :

Tentukan titik akhir dari ($x1$,$y1$) dan ($x2$,$y2$)

1) Jika x_1 = x_2 (garis vertikal), maka

- y = y+ 1 dan x tetap
- gambar titik (x,y) dilayar

2) Jika $y1 = y2$ (garis horisontal), maka

- $x = x + 1$ dan y tetap
- gambar titik (x,y) dilayar

{ anggap dari $x2 > x1$, (jika kebalikannya, maka $x2$ diganti $x1$)}

3) Hitunglah kemiringan dari titik $m = (y2 - y1)/(x2 - x1)$

4) $N = x2 - x1 + 1$

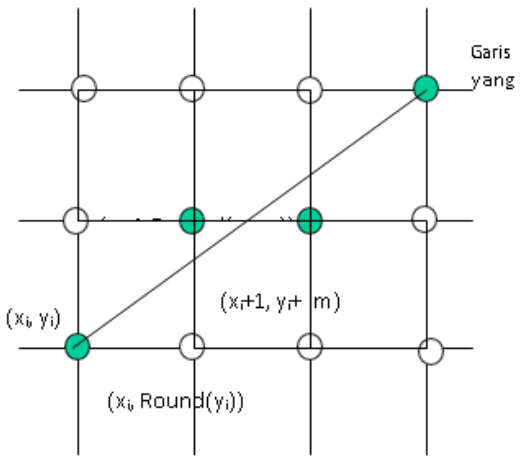
5) $x = x1$

6) aturan

- $y = m\,(x - x_1)$
- $y_a = Round\,(y)$,
- Buat gambar sesuai posisi (x, y_a)
- X = x + 1

7) Selesai

Karena posisi piksel berupa bilangan bulat (integer), Kemudian nilai *m* umumnya adalah pecahan, oleh karena itu *y* merupakan bilangan pecahan dan perlu dilakukan pembulatan nilai *y*.



Gambar 3. 5 Dibutuhkan pembulatan dari nilai **y** yang disepertemuankan letak titik piksel berupa bilangan bulat

Contoh 1:

Diketahui terdapat dua titik yaitu titik A yang terletak di kordinat (2,1) dan juga titik B (8,5), bila titik A dijadikan posisi awal kemudian titik B menjadi letak titik akhir, Dengan memakai algoritma brute force, coba gambarkan garis diantara kedua titik A dan B sehingga menghubungkan kedua titik tersebut.

Cara Menjawab :Posisi titik di awal ($x1$, $y1$) = (2,1), kemudian ($x2$, $y2$) = (8,5)

a. tidak terpenuhi

b. tidak terpenuhi

c. m = (5 – 1)/(8 – 2) = 0,67

d. N = 8 – 2 + 1 = 7

e. Kemudian terakhir adalah x = 2 y = 0,67 (x – 2) + 1

f. Ulangi langkah ini sampai 7 kali

Pengulangan ke-1:

x = 2; y = 0,67. (2 – 2) + 1 = 1

penggabungan ke nilai : y = 1, gambarlah pada posisi (2,1) dalam layar x = 2 + 1 = 3

Pengulangan ke-2:

x = 3; y = 0,67. (3 – 2) + 1 = 1,67

penggabungan ke nilai : y = 2 gambarlah pada posisi (3,2) dalam layar x = 3 + 1 = 4

Pengulangan ke-3:

x = 4; y = 0,67. (4 – 2) + 1 = 2,34

penggabungan ke nilai : y = 2 gambarlah pada posisi (4,2) dalam layar x = 4 + 1 = 5

Pengulangan ke-4:

x = 5; y = 0,67. (5 – 2) + 1 = 3,01

penggabungan ke nilai : y = 3 gambarlah pada posisi (5,3) dalam layar

$x = 5 + 1 = 6$

Pengulangan ke-5:

$x = 6$; $y = 0{,}67.\ (3 – 2) + 1 = 3{,}68$

penggabungan ke nilai : $y = 4$ gambarlah pada posisi (6,4) dalam layar $x = 6 + 1 = 7$

Pengulangan ke-6:

$x = 7$; $y = 0{,}67.\ (7 – 2) + 1 = 4{,}35$

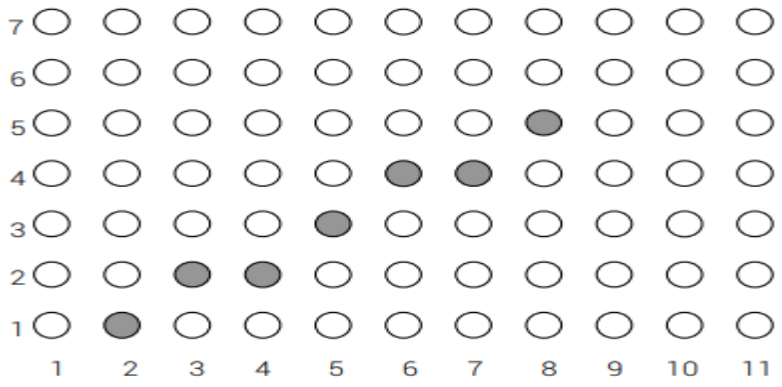
penggabungan ke nilai $y = 4$ gambarlah pada posisi (7,4) dalam layar $x = 7 + 1 = 8$

Pengulangan ke-7:

$x = 8$; $y = 0{,}67.\ (7 – 2) + 1 = 5{,}02$

penggabungan ke nilai : $y = 5$ gambarlah pada posisi (8,5) dalam layar , maka

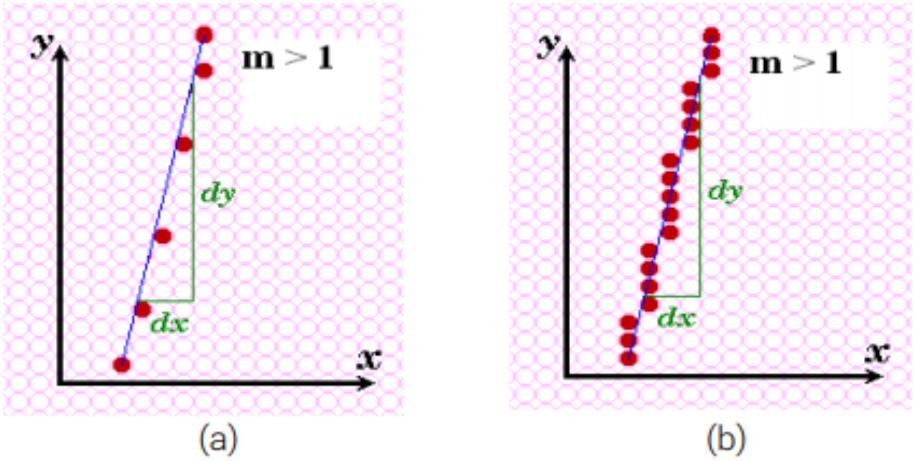
Akan didapatkan posisi titik-titik yang akan membentuk sebuah garis yaitu : (2,1), (3,2), (4,2), (5,3),(6,4), (7,4), (8,5).



Gambar 3. 6 Letak titik kordinat menggunakann perhitungan brute force algorithm dalam grafik raster.

Permasalahan :

Dalam hal gradien *m* > *1*, letak titik pembentuk dari garis terlihat tidak tersambung (gambar 3.7.a) hal ini menyepertemuankan adanya *gap*/jarak diantara piksel, sehingga perlu dibuat interpolasi diantara piksel.



Gambar 3. 7 (a) Gambar garis dengan kemiringan m > 1, terdapat jarak diantara titik (b) posisi setelah dilakukan interpolasi

Selain memakai teknik interpolasi, dalam hal kemiringgan m > 1, pindahkan x dengan y untuk menghilangkan jarak/gap yang ada diantara titik-titik tersebut. Selanjutnya rumus algoritma untuk garis m > 1 dapat dijelaskan seperti di bawah ini :

1) Tentukan kedua titik yang akan menjadi ujung masing-masing sumbu (x_1,y_1) dan (x_2,y_2)

2) Apabila $x_1 = x_2$ (garis vertikal), selanjutnya

   a) $y = y + 1$, kemudian x bernilai sama

   b) gambarlah letak posisi (x,y) dalam layar

   c) Langkah sudah selesai

3) Jika $y_1 = y_2$ (garis horisontal), selanjutnya

   a) $x = x + 1$, kemudian y bernilai sama

   b) gambarlah letak posisi (x,y) dalam layar

   c) Langkah sudah selesai

Misalkan nilai $y_2 > y_1$, (bila keballikannya, maka ganti y_2 sehingga menjadi y_1)

4) Hitunglah gradien dari garis $m = (x_2 - x_1) / (y_2 - y_1)$

5) $N = y_2 - y_1 + 1$

6) $y = y_1$

7) Ulangi sejumlah N pengulangan :

   a) $x = m\,(y - y_1) + x_1$

   b) Gabungkan nilai sampai mendekati x_a = round (x),

   c) gambarlah letak posisi (x_a,y) dalam layar

   d) $y = y + 1$

8) Semua langkah telah selesai.

Langkah tersebut merupakan salah satu contoh dalam mengggunakan Algoritma Brute Force, namun bukan berarrti algoritma ini tidak memiliki kelemahan. Salah satu kelemahan dari penggunaan akgoritma adalah prosesnya yang memakan waktu cukup lama. Hal ini dikarenakan langkah iterasi/ pengulangan memiliki perkalian antar angka pecahan (floating point), pembulatan dann lainnya sampai didapatkan angka yang berbentuk bilangan integer (bilangan bulat).

Contoh 2:

Terdapat 2 (dua) titik yaitu simbol A (4,3) dan juga simbol B (7,8). Diketahui A menjadi titik awal sedangkan letak posisi akhirnya adalah simbol B, dengan algoritma *brute force* buatlah gambar garis yang menghubungkan antara kedua titik tersebut.

Cara Menjawab :

a. titik awalan A (4,3), kemudian simbol B (7,8)

b. Belum terpenuhi

c. Belum terpenuhi

d. $m = (7 – 4)/(8 – 3) = 0,6$

e. N = 8 – 3 + 1 = 6

f. nilai y = 3, x = 0,6. (y – 3) + 4

g. Ulangi sejumlah 6 (enam) pengulangan

Pengulangan ke- 1:

y= 3; x = 0,6. (3 – 3) + 4 =4

bulatkan nilai mendekati : x = 4

buat gambar di posisi (4,3) dalam layar y = 3 + 1 = 4

Pengulangan ke- 2:

y= 4; x = 0,6. (4 – 3) + 4 =4,6

bulatkan nilai mendekati : x = 5

buat gambar di posisi (5,4) dalam layar y = 4 + 1 = 5

Pengulangan ke- 3:

y= 5; x = 0,6. (5 – 3) + 4 =5,2

bulatkan nilai mendekati : x = 5

buat gambar di posisi (5,5) dalam layar y = 5 + 1 = 6

Pengulangan ke- 4:

y= 6; x = 0,6. (6 – 3) + 4 =5,8

bulatkan nilai mendekati : x = 6

buat gambar di posisi (6,6) dalam layar y = 6 + 1 = 7

Pengulangan ke- 5:

y= 7; x = 0,6. (7 – 3) + 4 =6,4

bulatkan nilai mendekati : x = 6

buat gambar di posisi (6,7) dalam layar y = 7 + 1 = 8

Pengulangan ke- 6:

y= 8; x = 0,6. (8 – 3) + 4 =7

bulatkan nilai mendekati : x = 7

buat gambar di posisi (7,8) dalam layar $y = 8 + 1 = 9$

Maka akan didapatkan posisi pembentukan garis : (4,3), (5,4), (5,5), (6,6), (6,7), (7,8).

d. Algorithm DDA (Digital Differential Analyzer)

Algoritma DDA memiliki dasar persamaan yaitu (2 - 8). Setiap garis memakai persamaan awal (x1,y1) dan akhiran (x2,y2). Selanjutnya titik koordinat dilakukan penghitungan dan diubah sampai hasilnya menjadi bilangan bulat (integer). Salah satu kelebihan algoritma DDA yaitu dappat dipakai dalam menghitung dengan semua kemiringan garis ($0 < m < 1$; $m>1$; $-1 < m < 0$; $m < -1$). Dibawah ini merupakan cara-cara membuat garis dengan menggunakan algoritma Digital Differential Analyzer (DDA):

a. Pilihlah titik yang ingin dihubungkan dengan membentuk garis lurus.
b. Pilih salah satu titik yang akan menjadi awalan (x_1, y_1) sedangkan titik lainnya akan diposisikan menjadi letak posisi di akhir (x_2, y_2).
c. Hitunglah : $dx = x_2 - x_1$, selanjutnya $dy = y_2 - y_1$
d. Lanjutkan langkah, lakukan perhitungan menggunakan aturan sebagaimana dibawah ini :
   - jika $|dx| > |dy|$ artinya $step = |dx|$
   - jika bukan, artinya $step = |dy|$
e. Hitunglah penjumlahan piksel kordinat menggunakan perhitungan:
   - $x_inc = dx / step$
   - $y_inc = dy / step$
f. Kemudian lanjutkan : $x = x + x_inc$ $y = y + y_inc$
g. Hitung penambahan angka hingga mendekati $u = \text{Round}(x)$, $v = \text{Round}(x)$, selanjutnya letakan piksel (u, v) pada layar
h. Lakukan lagi langkah ke 6 & 7 dalam membantu mencari letak piksel lainnya hingga didapatkan $x = x_2$ dan $y = y_2$.

Contoh

Terdapat 2 letak titik yaitu A (2,1) selanjutnya yaitu B(8,5), Apabila simbol posisi A diketahui menjadi letak titik awal, kemudian letak posisi akhirnya adalah titik B, Cobalah untuk membuat garis yang

menghubungkan kedua titik dengan mmemakai algoritma Digital Differential Analyzer.

Cara Menjawab :

Titik awal (x_1, y_1) = A(2,1) dan Titik akhir (x_2, y_2) = B(8,5)

$dx = x_2 - x_1 = 8 - 2 = 6$ dan $dy = y_2 - y_1 = 5 - 1 = 4$

Karena: $|dx| > |dy|$, maka $step = |dx| = 6$

$x_inc = dx / step = 6/6 = 1$

$y_inc = dy / step = 4/6 = 0{,}67$

Pengulangan ke-1: $(x,y) = (2,1)$

$x+x_inc = 2 + 1 = 3$

$y+y_inc = 1 + 0{,}67 = 1{,}67$

Nilai kordinat berikutnya : $(x,y) = (3; 1{,}67)$

Angka yang mendekati (3; 1,67) ≈ (3,2). Gambarlah (3,2) dalam layar

Pengulangan ke-2: $(x,y) = (3; 1{,}67)$

$x+x_inc = 3 + 1 = 4$

$y+y_inc = 1{,}67 + 0{,}67 = 2{,}34$

Nilai kordinat berikutnya: $(x,y) = (4; 2{,}34)$

Angka yang mendekati (4; 2,34) ≈ (4,2). Gambarlah (4,2) dalam layar

Pengulangan ke-3: $(x,y) = (4; 2{,}34)$

$x_A+x_inc = 4 + 1 = 5$

$y_A+y_inc = 2{,}34 + 0{,}67 = 3{,}01$

Nilai kordinat berikutnya: $(x,y) = (5; 3{,}01)$

Angka yang mendekati (5; 3,01) ≈ (5,3). Gambarlah (5,3) dalam layar

Pengulangan ke-4: $(x,y) = (5; 3{,}01)$

$x_A+x_inc = 5 + 1 = 6$

y_A+y_inc = 3,01 + 0,67 = 3,68

Nilai kordinat berikutnya: (*x,y*) = (6; 3,68)

Angka yang mendekati (6; 3,68) ≈ (6,4). Gambarlah (6,4) dalam layar

Pengulangan ke-5: (*x,y*) = (6; 3,68)

x_A+x_inc = 6 + 1 = 7

y_A+y_inc = 3,68 + 0,67 = 4,35

Nilai kordinat berikutnya: (*x,y*) = (7; 4,35)

Angka yang mendekati (7; 4,35) ≈ (7,4). Gambarlah (7,4) dalam layar

Pengulangan ke-6: (*x,y*) = (7; 4,35)

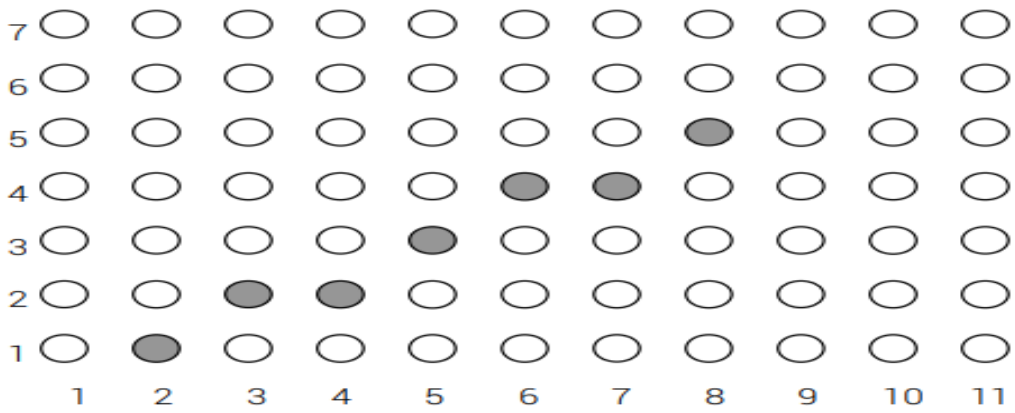
x_A+x_inc = 7 + 1 = 8

y_A+y_inc = 4,35 + 0,67 = 5,02

Nilai kordinat berikutnya: (*x,y*) = (8; 5,02)]

Angka yang mendekati (8; 5,02) ≈ (8,5). Gambarlah (8,5) dalam layar Setelah didapatkan $x = x_2 = 8$, maka pengulangan tidak diperlukan lagi, Maka didapatkan hasil akhir posisi titik yaitu: (2,1), (3,2), (4,2), (5,3),

(6,4), (7,4) dan (8,5).

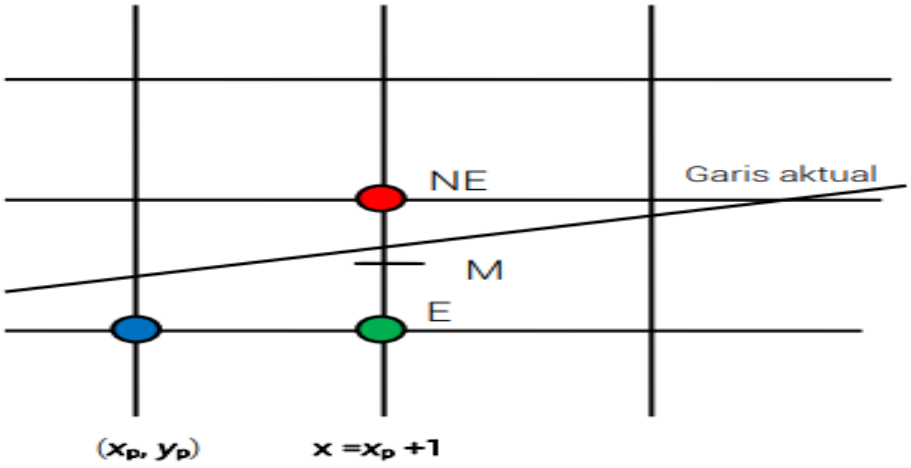


Gambar 3. 8 Gambar raster graphics pemmbentukan garis memakai hasil perhitungan algoritma Digital Differential Analyzer.

Algoritma Digital Differential Analyzer (DDA) memiliki kelebihan jika dibandingkan dengan Algoritma Brute Force. Kelebihan tersebut terletak pada prosesnya yang menghabiskan waktu raltif lebih cepat dan juga lebih baik saat digunakan dengan kemiringan garis m>1. Akan tetapi algoritma DDA juga memiliki kelemahan yaitu prosedur dalam membuat garis tetap diharuskan melakukan pembulatan. Kekurangan lainnya adalah karena variabel *x, y* dan juga *m* tetap mengharuskan angka bilangan real dan bukan bilangan pecahan.

2. Algoritma Bresenham

Algoritma Bresenham pertama kali dicetuskan di tahun 1965 oleh J.E. Bresenham. Temuannya berhasil mengoptimalkan kualitas gambar garis khususnya di dalam *raster graphics*. Algoritma ini sudah tidak memakai *floating point arithmethic*. Dalam algoritma Bresenham, angka pembulatan pada posisi piksel tidak lagi diperlukan. Sehingga iterasi (pengulangan) yang dilakukan cukup memakai sistem *incremental* saja. Kelebihan lainnya adalah karena algoritma ini cukup kompatibel dengan sotware maupun hardware yang digunakan untuk menggambar garis. Kekurangannya adalah karena algoritma ini hanya bisa digunakan untuk posisi kemiringan (0 < m < 1) saja.



Gambar 3. 9 Gambar garis yang terletak antara titik NE dan E dan memiliki garis tengah M.

Perhatikan Gambar 3.9 apabila gradien pada garis *m,* yang yang memiliki nilai 0 < *m* < 1, kemudiaan memiliki posisi awalan pada (x_0, y_0) sedangkan posisi

akhirnya berada pada (*x1,y1*). Kita ibaratkan letak simbol/titik saat ini berada di titik (x_p, y_p) sehingga harus ditentukan titik mana yang akan menjadi posisi simbol *East* (E) dan juga *Northeast* (NE). Jika simbol Q merupakan perpotongan antara garis yang memiliki nilia garis $x = x_p+1$, kemudian simbol M merupakan posisi tengah yang terletak antara simbol NE dan jug E. Apabila simbol M yang menjadi letak titik tengah berada di atas dari garis, maka simbol E yang akan dipilih, sebaliknya yaitu apabila simbol M berada di bawah garis, berarti simbol NE yang akan dipilih. Untuk kasus seperti ini sebaiknya ditentukan dimana letak garis, apakah akan terletak diatas titik M atau berada dibawahnya dari titik M tersebut. Dalam menggunakan algoritma ini, hal yang perlu diingat adalah setiap garis dapat didefinisikan seperti penjelasan dibawah ini:

$$F(x, y) = ax + by + c = 0, \text{ Terbentuklah nilai } 3 - 1$$

Hitunglah: $dy = y_1 - y_0$ dan $dx = x_1 - x_0$, berarti dapat kita tulis seperti berikut :

$$y = mx + B = (dy/dx)\, x + B,\ (dy/dx)\, x - y + B = 0$$

Lakukan penghitungan perkalian menggunakan d*x* sehingga (d*y*) – (d*x*)*y* + (d*x*)B = 0 Dan didapatkan $a = dy$, $b = -dx$ dan $c = (dx)B$.

Apabila F (*x, y*) = 0, maka (*x, y*) memiliki letak tepat dia garis

Apabila F (*x, y*) > 0, maka (*x, y*) memiliki letak di bawah dari garis

Apabila F (*x, y*) < 0, maka (*x, y*) memiliki letak di atas dari garis

Dalam menggunakan karakteristik midpoint, maka coba hitung:

$F(M) = F(x_p + 1, y_p + ½) = a(x_p + 1) + b(y_p + ½) + c$ Kemudian ujilah simbol dari $d = F(M)$.

Jika $d > 0$, M memiliki posisi di bawah dari garis, artinya kita pilih NE. Jika $d = 0$, ambil E (walaupun kita dapat mengambil pilihan NE)

Jika $d < 0$, M memiliki posisi diatas garis, oleh karena itu kita pilih E.

Karena kita mengambil E, maka langkah M ditambah satu kearah sumbu x. sehingga diperoleh d terbaru yaitu:

$$\mathbf{d}_{baru} = F(x_p + 2, y_p + ½) = a\,(x_p + 2) + b\,(y_p + ½) + c$$

$$\mathbf{d}_{lama} = F(x_p + 1, y_p + ½) = a\,(x_p + 1) + b\,(y_p + ½) + c$$

jadi $\mathbf{d}_{baru} - \mathbf{d}_{lama} = a = dy.$

Setelah kita memilih E, penjumlahan dalam *d* baru yaitu *a* = *dy*. Apabila kita memilih NE, maka lakukan penambahan M menuju ke *x* dan *y,* maka akan didapatkan :

$$d_{baru} = F(x_p + 2, y_p + 3/2) = a\,(x_p + 2) + b\,(y_p + 3/2) + c$$

$$d_{lama} = F(x_p + 1, y_p + ½) = a\,(x_p + 1) + b\,(y_p + ½) + c$$

jadi $d_{baru} - d_{lama} = a + b = dy - dx.$

Ketika algoritma dimulai maka persamaan awalnya adalah sebagai berikut (x_p , y_p) = (x_0, y_0). Dalam menjumlahkan nilai d di awal algoritma, maka langkah midpoint nya adalah (x_0 + 1, y_0 + ½) dan

$$F(M) = F(x_0 + 1, y_0 + ½) = a\,(x_0 + 1) + b\,(y_0 + ½) + c$$

$$= ax_0 + by_0 + c + a + b/2$$

$$= 0 + a + b/2 \quad \text{karena } (x_0, y_0) \text{ terletak pada garis.}$$

Jadi kita mulai dari $d = a + b/2 = dy - dx/2$

Supaya angka tersebut hilang, lakukan perhitungan F, caranya adalah melakukan perkalian 2, sehingga didapatkan F(*x,y*) = 2 (*ax* + *by* + *c*) = 0. Perhitungan ini membuat nilainya menjadi *d* = 2*dy* – *dx* Sehingga langkah-langkah memakai algortima bresenham midpoint (dengan nilai kemiringan 0 < m < 1) :

a. Pilihlah titik-titik yang ingin dihubungkan dalam membentuk sebuah garis.
b. Pilihlah dimana letak posisi awal (x_0, y_0) juga letak posisi akhir (x_1, y_1).
c. Hitunglah *dx*, *dy*, 2|*dy*| dan 2|*dy*| − 2|*dx*|
d. Jumlahkan parameter : *p*o = 2|*dy*| − |*dx*|
e. Dalam setiap *xk* sejajar dengan garis, kita mulai menggunakan k = 0

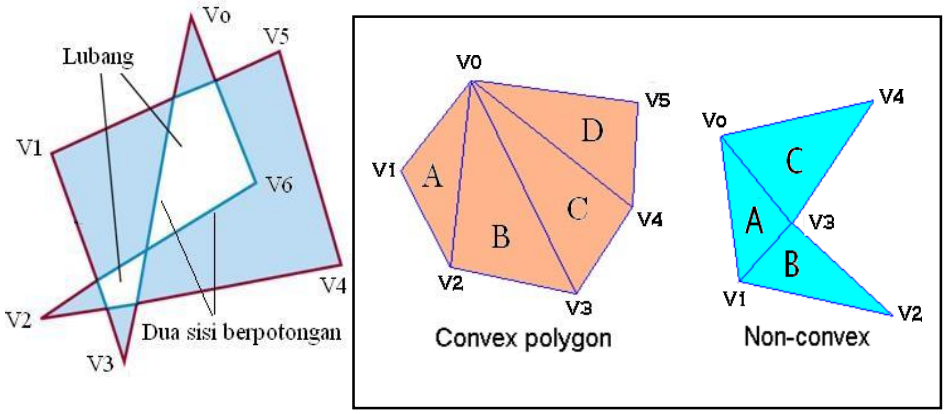
- jika $p_k < 0$ maka letak berikutnya yaitu : (x_k+1, y_k) dan $p_{k+1} = p_k + 2|dy|$
- jika bukan, maka posisi titik berikutnya yaitu :

$$(x_k+1, y_k+1) \text{ dan } p_{k+1} = p_k + 2|dy| - 2|dx|$$

f. Ulang langkah ke- 5 dalam mendapatkan dimana letak pixel selanjutnya hingga didapatkan $x = x_1$ dan $y = y_1$.

3. Switch

. Bentuk poligon merupakan gabungan dari beberapa garis lurus yang masing-masing terhubung sampai menjadi sebuah bentuk luasan (bangun luas). Garis di dalam bentuk poligon dikenal dengan nama edge (garis sisi poligon). Selain edge, dalam bentuk poligon terdapa pula yang disebut verteks yang merupakan titik dari pertemuan dua sisi setiap garis. Bentuk poligin umumnya disebut dengan kordinat verteks. Secara umum dikenal dua jenis poligon yaitu, poligon sederhana dan juga poligon tidak sederhana. Masing-masing poligon memiliki cirinya tersendiri. Untuk poligon sederhana memiliki ciri yaitu tidak semua letak posisi verteksnya ada di dalam bidang yang serupa, kemudan ciri lainnya adalah tidak memiliki sisi yang berpotongan, serta tidak memiliki lubang dalam bentuknya.



Gambar 3. 10 Poligon berbentuk sederhana (a) dan poligon bentuk tidak sederhana (b)

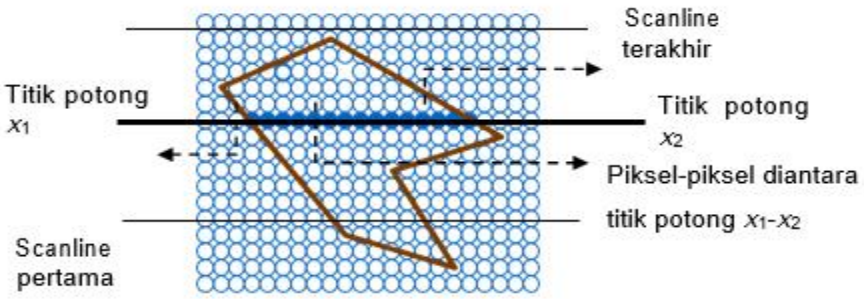
Secara umum poligon dapat dibedakan ke dalam dua bagian, pertama convex poligon dan kedua yaitu non convex poligon. Kedua poligon ini memiliki cirinya masing-masing. Untuk convex poligon bisa dilihat bahwa sudut interior poligonnya kurang dari 180 derajat, bisa dikatakan bahwa segmen yang dibuat

dari dua verteks berbentuk sembarang terletak di dalam poligon. Bangun poligon memiliki bentu sederhana yaitu segitiga, hal ini dikarenakan setiap poligon dapat dibuat menjadi bagian kecil yang akan berbentuk segitiga seperti yang terlihat dalam gambar 3.12 (b). Dalam prakteknya kita dapat membuat bentuk permukaan semua objek 3D apabila kita memiliki poligon dalam jumlah yang memadai.

Dalam dunia grafik, penambahan warna dilakukan sebagai upaya memperindah bentuk dan tampilan dari poligon. Maka, untuk melakukan hal tersebut dibutuhkan teknik algortima tersendiri dengan tujuan memberi warna did dalam poligon. Penambahan warna di dalam poligon disebut dengan teknik *area filling* atau *filling poligon*. Terdapat dua cara yang umum digunakan dalam melakukan *filling poligon* dalam sistem raster. Kedua teknik tersebut adalah *Boundary Fill Algorithm* dan *Scan Line Poligon Fill Algorithm.* Kita akan membahas penggunaan Scan Line Poligon Fill Algorithm terlebih dahulu.

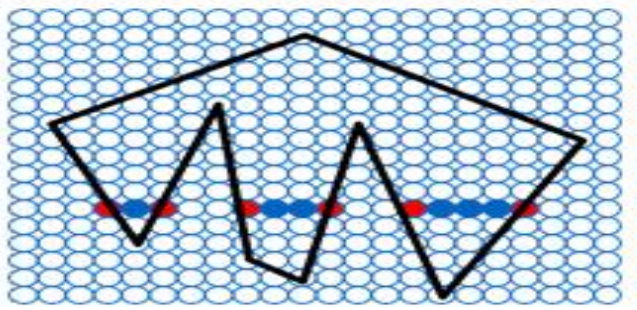
a. *Scan Line Poligon Fill Algorithms*

Cara ini dilakukan dengan penambahan warna dalam poligon melalui scan dengan arah horizontal dari sisi kiri ke arah sisi kanan. Teknik ini dipakai supaya didapatkan titik potong yang tepat dengan bagian pinggir poligon. Selanjutnya adalah mengurutkan titik perpotongan x sisi kiri menuju sisi kanan untuk selanjutnya diberikan pewarnaan di pixel yang ada antara kedua poligon yang bersisian atau ($x1$- $x2$). Teknik *scan line* dimulai dengan garis terletak di bagian bawah yang memiliki nilai *y* yang nilainya kecil sampai ke bagian garis scan yang terletak di bagian atas. Teknik ini dapat dipakai untuk melakukan penambahan warna dalam objek-objek sederhana misalnya elips, lingkaran dan bentuk sederhana lainnya. Seperti terlihat pada gambar berikut :



Gambar 3. 11 ilustrasi penggunaan metode scan line

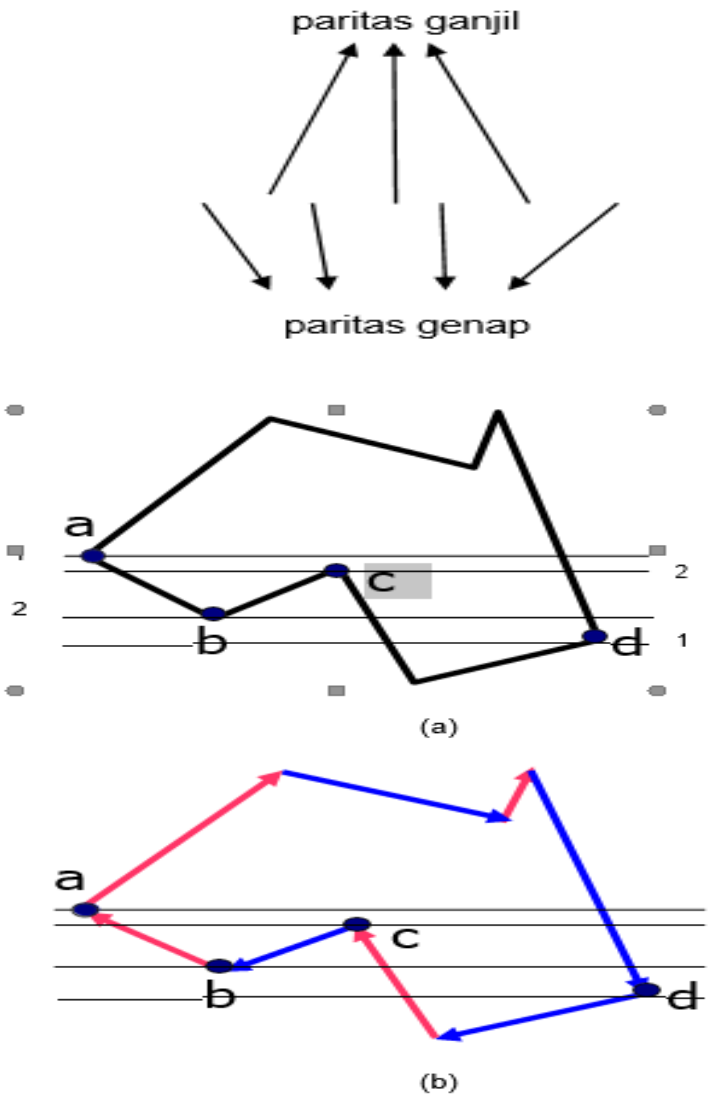
Permasalahan umum yang sering terjadi adalah bagaimana kita bisa menentukan apakah piksel tersebut ada di dalam atau di luar poligon. Salah satu caranya adalah dengan menggunakan *Inside-Outside test.* Dalam membedakan antara bagian sisi dalam dengan sisi luar, kita dapat menggunakanj aturan paritas ganjil genap (*odd even rule*). Langkah awal yang dilakukan adalah menngatur paritas dengan nilai paritas genap. Apabila kita menemukan titik potong dimana nilai paritas yang dibalik, maka yang semula nilainya genap kita balik sehingga menjadi nilai ganjil begitupula dengan sebaliknya. Kemudian adalah kita memberi warna piksel apabila nilai paritasnya adalah ganjil.



Gambar 3. 12 Ilustrasi penggunaan aturan paritas dan pengisian warna

Permasalahan lainnya antara lain bisa kita lihat pada gambar 3.15. Dapat dilihat bahwa verteks dari a, kemudian b, c dan juga d yang menjadi titik bertemunya kedua segmen garis tersebut. Pertanyaannya adalah

kenapa verteks a dan juga d hanya dilakukan penghitungan sebanyak satu kali saja, berbeda dengan verteks b dan c yang dilakukan dengan 2 kali penghitungan ?

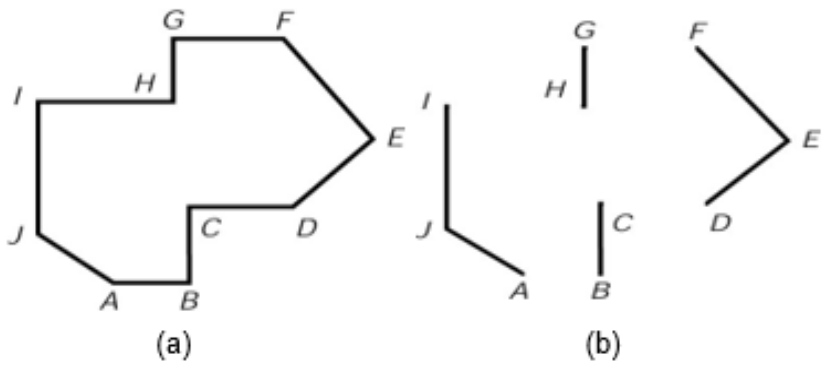


Gambar 3. 13 ilustrasi penghitungan aturan paritas dalam gambar verteks.

Jalan keluar untuk mengatasi hal ini dapati dilakukan cara mengubah arah putaran bagian tepi poligon menjadi searah dengan jarum jam, yang dapat dilihat di gambar 3.15.(b). Periksalah apabila terdapat verteks yang arah panahnya mengalami perubahan (dari mengarah keatas menjadi

kebawah, begitupula sebaliknya) mka penghitungan dalam verteks itu kita hitung sebanyak dua kali hitungan. Namun, apabila tidak terdapat perubahan arah panah pada verteks, artinya verteks tersebut hanya dihitung sekali saja.

Permasalahan lain yang umum terjadi adalah bagaimana cara melakukan perhitungan di garis tepi horizontal. Seperti yang terlihat dalam gambar 3.16.(a) Bagaimana cara menghitung bagian pinggir horizontal AB, CD, GF, dan juga HI ?



Gambar 3. 14 Bentuk poligon dan bagian sis horizontal

Cara menyelesaikan perhitungan tersebut adalah dengan mengabaikan verteks yang memiliki letak di pinggir dari horizontal dan jangan dimasukkan sebagai bagian dari perhitungan aturan paritas, seperti dapat dilihat pada gambar 3.16.(b).

Algoritma pengisian warna scan line memakai aturan paritas ganjil genap sebagai berikut :

1) Pilihlah titik potong garis dengan semua bagian dari poligon
2) Atur dan urutkan semua titik dengan mendasarkan pada kordinat sumbu x
3) Beri warna pada semua piksel yang ada pada pasangan titik (x1-x2) yang posisinya berada di dalam poligon dengan menggunakan aturan paritas ganjil – genap.

Akan tetapi algoritma *scan fill* juga memiliki kelemahan. Kelemahan dari algoritma *scan fill* adalah diperlukan biaya yang besar, hal ini dikarenakan semua sisi poligon perlu dilakukan pengjujian khususnya terhadap masing-masing piksel.

Salah satu jalan keluar untuk mengatasi kelemahan tersebut adalah dengan penggunaan *edge table concept*. *Edge Table* (ET) sendiri merupakan sebuah tabel berisi sisi- sisi dari poligon. Selain itu, terdapat juga *Active Edge Table* (AET) dan juga *Global Edge Table* (GET). Kedua edge table ini memiliki fungsi yang berbeda, AET memiliki fungsi menyimpan seluruh bagian sisi dari poligon yang memiliki perpotongan dengan garis scan. Kemudian, untuk GET fungsinya adalah menyimpan seluruh bagian sisi dari poligon dan melakukan update dari AET.

Cara perhitungan garis scan dan titik potongnya :

Persamaan Umum :

$$y = mx + b, \quad m = \frac{y_{\max} - y_{\min}}{x_{@y\max} - x_{@y\min}}$$

Semua *scan line* baru maka ketentuan yang digunakan yaitu :

$$y_{i\square 1} \square y_{i} \square 1$$

Perhitungan x kepada semua *scan line* :

$$x = \frac{y - b}{m}$$

.

Sehingga didapatkan persamaan seperti berikut :

$$x_i = \frac{y_i - b}{m}, \qquad x_{i+1} = \frac{y_{i+1} - b}{m}$$

$$y_{i+1} = y_i + 1$$

Maka hasil akhirnya adalah :

$$x_{i+1} = \frac{y_i + 1 - b}{m} = \frac{y_i - b}{m} + \frac{1}{m} = x_i + \frac{1}{m}$$

Dimana hal ini merupakan salah satu teknik yang memiliki efisiensi yang baik dalam melakukan perhitungan terhadap nilai *x.*

1) Penjelasan AET
   a) Merupakan tabel yang mengandung setiap entry dari sisi yang berpotongan dengan *scan line* yang aktif.
   b) Untuk semua *scan line* baru maka :
      (1) Lakukan perhitungan perpotongan baru pada seluruh sisi dengan memakai rumus :
      (2) Menambahkan seluruh sisi yang baru terbentuk dan berpotongan.
      (3) Kemudian hilangkan/hapus seluruh bagian sisi yang tidak memiliki perpotongan.
      (4) Dalam memaksimalkan efisiensi dalam update Active Entry Table, maka kita harus menjaga Global Entry Table.
2) Penjelasan GET
   a) GET merupakan tabel yang memuat informasi mengenai seluruh bagian sisi dari poligon.
   b) GET memiliki sebuah tempat untuk tiap-tiap *scan line*.
      (1) Seluruh tabel mempunyai daftar setiap bagian sisi dan juga memiliki nilai ymin
      (2) Tiap-tiap bagian sisi bisa ditempatkan ke dalam satu tempat.
   c) Setiap entry untuk di GET memiliki :
      (1) Angka y_{max} di setiap bagian sisi
      (2) Angka x@ymin (angka x dalam letak posisi y_{min})
      (3) Angka kenaikan dari x (1/m)

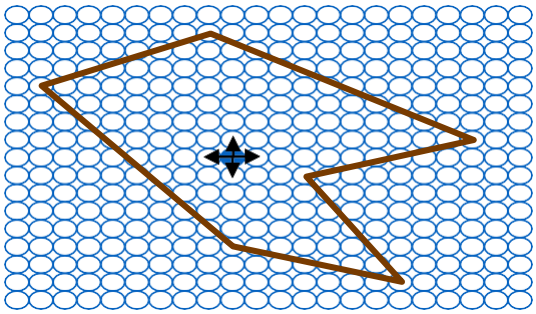
Penggunaan teknik ***Scan Line Poligon Fill Algorithms*** memakai bantuan GET, ET dan juga AET. Scan

1) Jumlahkan bagian sisi dari poligon ke dalam GET
2) Aturlah *y* ke kordinat *y* yang paling kecil ke arah GET
3) Aturlah set AET = habis
4) Ulangi langkah tersebut sampa AET dan GET benar-benar kosong
   a) Jumlahkan bagian sisi GET ke dalam AET dimana $y_{min} = y$.
   b) Hapus bagian sisi AET jika $y_{max} = y$.
   c) Susun secara berurutan AET dengan berdasar kepada *x*
   d) Beri warna pada piksel yang posisinya berada diantara titik potong yang terdapat di AET.
   e) Disetiap bagian sisi dalam dari AET, ubah nilai x menjadi x = 1/m
   f) Aturlah $y = y + 1$ supaya berpindah ke bagian line selanjutnya.

b. Boundary-Fill Algorithm

Penggunaan teknik Boundary Fill Algorithm memiliki prosedur dengan 3 parameter utama. Parameter tersebut antara lain kordinat dari letak (x,y), kemudian pewarnaan pengisian kemudian batas garis warna. Jadi, penambahan warna dalam poligon diawali di kordinat letak (x,y), selanjutnya adalah mengecek letak dari titik tetangganya, apakah terdapat warna batas atau tidak. Apabila tidak terdapat warna batas, maka beri warna titik itu dengan menggunakan warna tertentu. Kemudian, cek kembali letak dan warna titik tetangganya. Ulangi proses itu sampai semua titik dalam wilayah pengisian sudah selesai diuji semuanya.

Dalam teknik Boundary Fill, proses pengisian warna diawali dengan sebuah titik yang terletak di dalam wilayah (x,y) poligon kemudian beri warna piksel hingga seluruh piksel di dalam bentuk poligon selesai diberikan warna, kita dapat melihatnya dalam gambar 3.15.

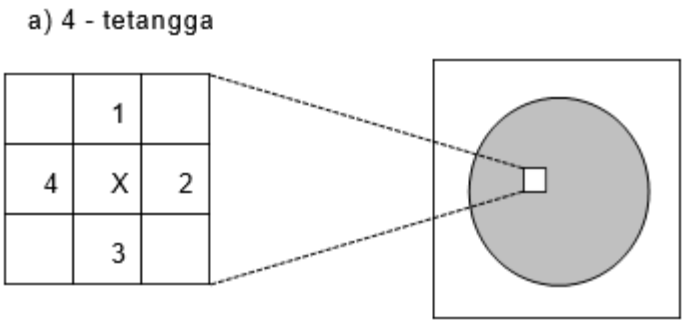


Gambar 3. 15 Ilustrasi penggunaan Boundary Fill pada raster

Untuk mengetahui dimana letak titik tetangga berada, terdapat 2 (dua) macam cara dalam melihat letak dari titik tetangga tersebut, yaitu :

1) Mencari titik yang memiliki posisi di atas, sebelah kanan, sebelah kiri dan bagian bawah dari titik pusat, 4 titik tetangga.
2) Mencari titik yang memiliki posisi di atas, kanan, kiri, pojok kiri atas, pokok kanan atas, pojok kanan bawah, pojok kiri bawah dan bagian bawah dari titik pusat, 8 titik tetangga.
3) Mencari letak posisi yang terletak diatas, kanan, bawah, kiri, pojok kanan atas, pojok kiri atas, pojok kanan bawah dan pojok kiri bawah.

Contohnya bisa terlihat seperti gambar 3.18 :



b)8-tetangga

1	2	3
8	X	4
7	6	5

Gambar 3. 16 Ilustrasi bentuk titik tetangga, 4 titik dan 8 titik tetangga

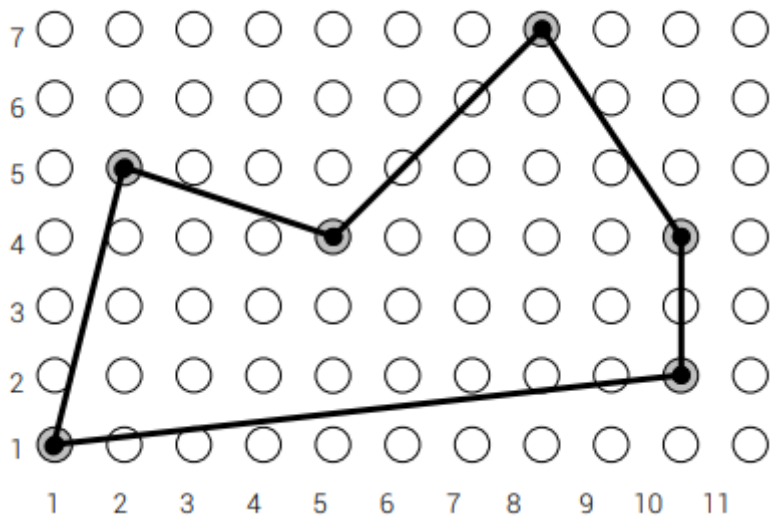
Contoh 1:

Contoh lainnya, Terdapat sebuah poligon dengan titik = (1,1), (2,5), (5,4), (8,7), (10,4), (10,2) dan (1,1), dengan memakai algoritma Boundary Fill 4-Connected, lakukan pengisian warna terhadap poligon tersebut,

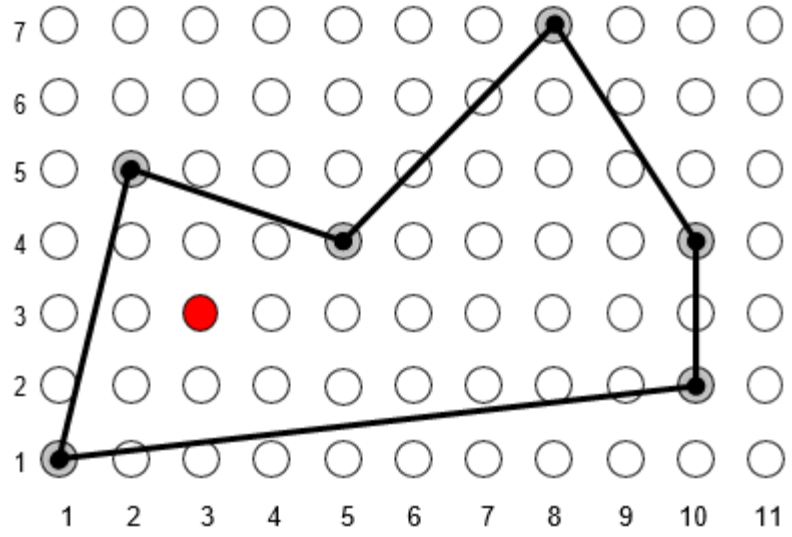
Cara Menjawab :

Diketahui titik pembentuk bangun poligon = {(1,1), (2,,5), (5,4), (8,7), (10,4), (10,2), (1,1)}.

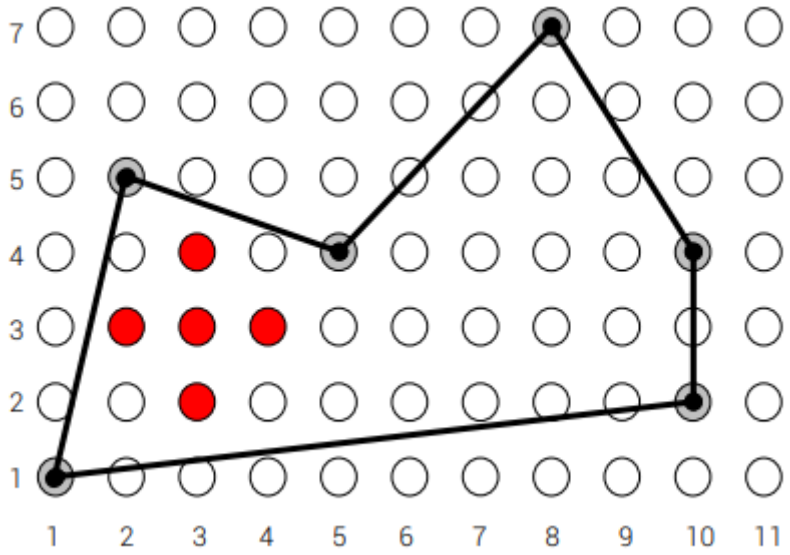
Jika dilakukan penggambaran, maka akan didapatkan gambar poligon seperti berikut ini :



Jika titik awalan yaitu (3,3), maka berikan tanda di posisi kordinat (3,3) menggunakan warna merah sebagai contohnya. Selanjutnya lihat ke 4 bagian titik tetangganya (3,4), (3,2), (4,3), (2,3) seperti berikut ini



Keempat titik itu bukanlah garis batas dari poligon, maka keempat titik itu diberi warna merah.



Letak titik yang sudah selesai mengalami pemrosesan : (3,3)

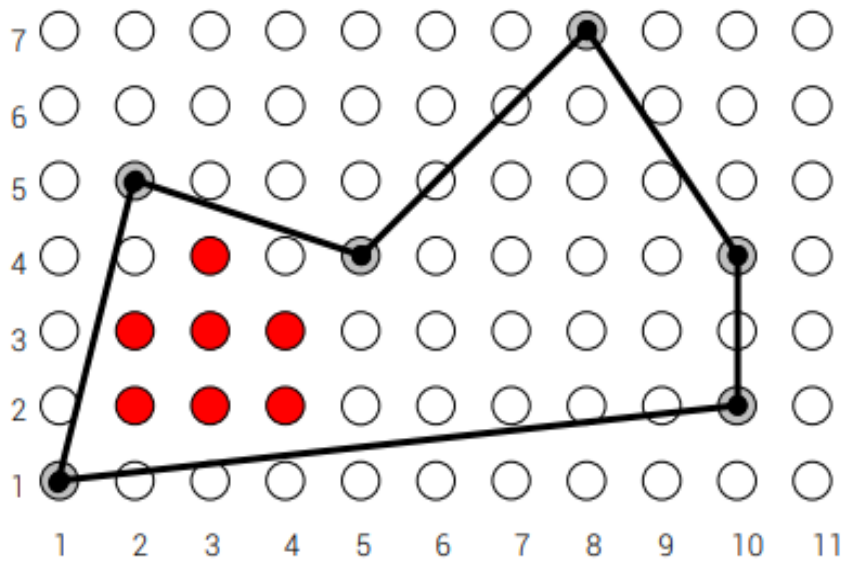
Letak titik belum mengalami pemrosesan : (3,2), (3,4), (2,3), (4,3)

Pilih titik (3,2).

Letak titik sudah selesai mengalami pemrosesan: (3,2), (3,3)

Letak titik belum mengalami pemrosesan : (3,4), (2,3), (4,3)

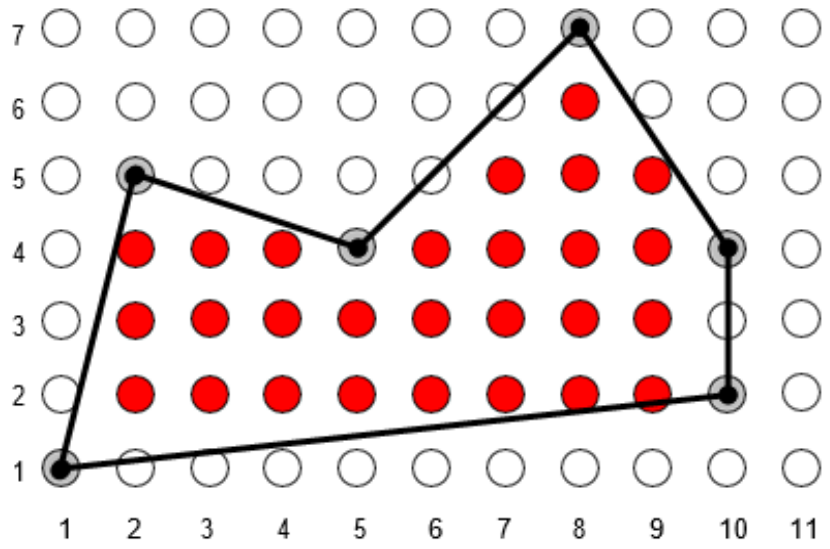
Keempat letak kordinat itu antara lain (3,3), (3,1), (2,2), (4,2). Dapat dilihat di kordinat (4,2) kemudian titik (2,2) bukanlah merupakan titik yang menjadi batas dari poligon. Maka titik tersebut juga diberikan pewarnaan merah. Selanjutnya kordinatk (3,3) sudah diberikan warna, kordinat (3,1) yang menjadi pembatas dari poligon tersebut tidaklah perlu diberikan warna.



Letak titik yang sudah selesai mengalami pemrosesan : (3,2)(3,3)

Letak titik belum mengalami pemrosesan : (2,3), (3,4), (4,2) (2,2), (4,3)

Pilih titik (3,4). Keempat titik tetangga tersebut terletak pada titik (3,5), (3,3), (4,4), (2,4). Titik (3,3) diberikan warna. Sedangkan titik yang terletak di (2,4), (4,4) dan (3,5) menjadi garis batas sehingga tidak diberikan warna. Ulangi proses tersebut sampai semua bagian di dalam bangun poligon diberi warna merah,


7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10 11

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 3	1. Terdapat dua simbol yaitu simbol A dan simbol B. Jika simbol A dijadikan sebagai posisi awalan dan simbol B menjadi posisi akhir, cobalah buat titik penghubung diantara simbol A dan simbol B sampai menjadi sebuah garis AB memakai algoritma : a. Brute Force Algorithm b. Digital Differential Analyzer Algorithm c. Bresenham Algorithm Dengan menggunakan ketentuan sebagai berikut : Titik A terletak pada kordinat (3,2) Titik B terletak pada kordinat (11,6)

D. REFERENSI

PERTEMUAN 4

ALGORITMA LINGKARAN DAN ELIPS

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan algortima lingkaran dan elips.

Pada pertemuan ini akan dijelaskan mengenai :

1. Algoritma Pembuatan Lingkaran
2. Algoritma Elips

B. URAIAN MATERI

1. Algoritma Pembuatan Lingkaran

Secara umum lingkaran merupakan kumpulan titik-titik yang menyebar serta mempunyai jarak dengan titik pusat yang serupa pada semua titik yang tersebar tersebut. Bangun lingkaran umumnya terbentuk dengan cara menggambar seperempat lingkaran yang sama simetris dengan bagian lainnya. Sehingga bertambahnya x bisa dilakukan dari nol ke r dengan menambahkan sejumlah bagian *step*, yaitu dengan menjumlahkan ± y kepada tiap-tiap *step*.

Persamaan lingkaran yang memiliki titik pusat di koordinat(*x*p, *y*p) kemudian jari-jari *r* dinotasikan dengan persamaan berikut:

$$(x - x_p)^2 + (y - y_p)^2 = r^2$$

Persamaan 3 - 2

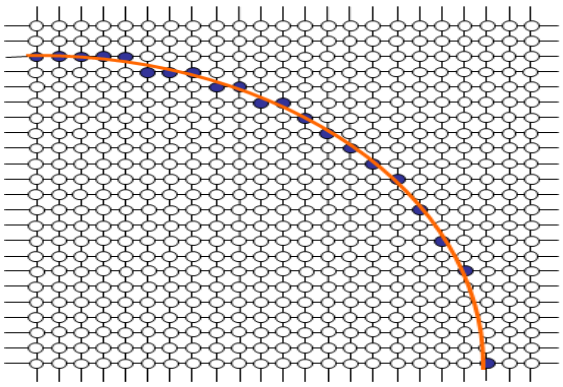
Atau

$$y = y_p \pm \sqrt{[r^2 - (x - x_p)^2]}$$

Persamaan 3 - 3

Dalam melakuka penggambaran piksel-piksel di dalam kurva sebuah lingkaran, maka kita bisa memakai sumbu x = (*x*p-*r*) hingga *x* = (*x*p+*r)* untuk menjadi acuan, kemudian menggunakan sumbu y yang menjadi hasil persamaan kordinat (2-10). Salah satu kesulitan dalam menggunakan algoritma

ini adalah karena diperlukan waktu operasi yang cukup besar, hal ini disepertemuankan karena algoritma ini membutuhkan operasi perkalian dua bilangan bulat (integer). Sehingga dibutuhkan fungsi yang mendukung yaitu SQRT (fungsi pendukung penghitungan akar bilangan) kemudian adalah fungsi ROUND (fungsi pendukung dalam membulatkan bilangan pecahan menjadi bilangan bulat/integer). Selain itu kesulitan lainnya adalah kordinat piksel yang dihasilkan tidak merata hal ini dikarenakan adanya jarak/gap yang disepertemuankan dari perubahan gradien.



$$y_0 = \sqrt{20^2 - 0^2} \approx 20$$
$$y_1 = \sqrt{20^2 - 1^2} \approx 20$$
$$y_2 = \sqrt{20^2 - 2^2} \approx 20$$
$$\vdots$$
$$y_{19} = \sqrt{20^2 - 19^2} \approx 6$$
$$y_{20} = \sqrt{20^2 - 20^2} \approx 0$$

Gambar 4. 1 Adanya jarak/gap dikarenakan perubahan gradien sehingga menghasilkan kordinat pada piksel menjadi tidak merata.

Untuk mengurangi resiko letak persebaran kordinat pixel tidak seimbang, kordinat letak pixel (*x,y*) diubah menggunakan kordinat polar dengan menggunakan persamaan (3 - 4)

$x = x_p + r\cos\theta$ dan $y = y_p + r\sin\theta$

Menggunakan persamaan 3 – 4

Walaupun begitu, cara ini membutuhkan waktu pengoperasian cukup besar dikarenakan adanya perhitungan angka real, penjumlahan trigonometri, kemudian juga memerlukan segmen bagian dari garis yang cukup banyak.

a. Simetris Delapan Titik (lingkaran)

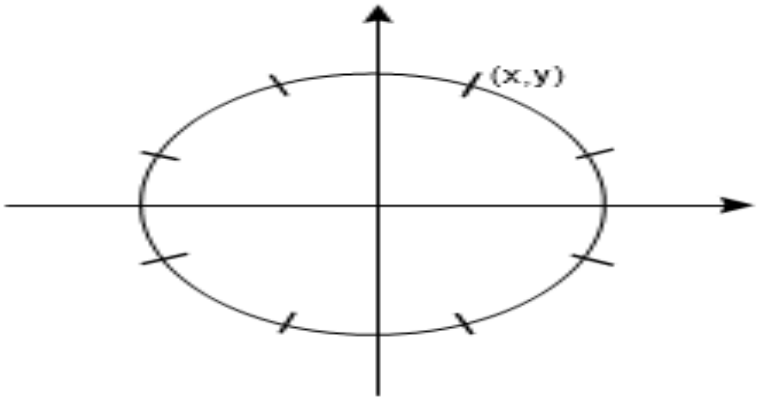
Penggambaran lingkaran kurva bisa dimulai dari penentuan dimana letak titik awal (x,y) yang ada di lingkaran, sehingga titik-titik lainnya dapat kita tentukan pula dengan menggunakan ketentuan berikut :

(−x,y), (x, −y), (−x, −y), (y,x), (−y,x), (y, −x), (−y, −x)

Sampai pada akhirnya akan menjadi 8 buah kordinat :

(x, y), (−x,y), (x, −y), (−x, −y), (y,x), (−y,x), (y, −x), (−y, −x)

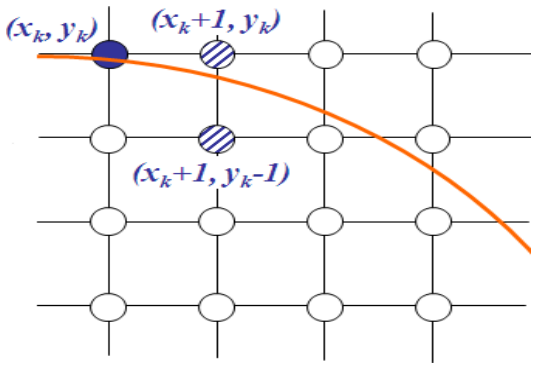
Jadi, hal ini diperlukan dalam menghitung segmen 45 derajat untuk melengkapi lingkaran sampai terbentuk sempurna.



Gambar 4. 2 Bentuk lingkaran simetris delapan titik

b. Midpoint Algorithm

Midpoint algorithm dikenal juga dengan nama algoritma lingkarab bresenham. Pembuatan titik pada algoritma ini didasarkan kepada titik pusat lingkaran dengan penambahan di seluruh jalur yang ada di sekeliling lingkaran. Langkah pertama komputasi dimulai dengan mengidentifikasi bagian yang ada di lingkaran dengan memakai sifat dari simetri. Caranya adalah dengan membagi lingaran sehingga masing-masing bagian memiliki sudut sebesar 45 derajat, jadi lingkaran tersebut akan terbagi menjadi 8 bagian yang sama besar.



Gambar 4. 3 ilustrasi letak garis yang terletak di titik antara (xk+1, yk) juga (xk+1, yk − 1)

Misalnya kita memiliki letak kordinat (*xk, yk*). Langkah berikutnya yaitu menentukan kordinat mana diantara (*xk*+1, *yk*) dan (*xk*+1, *yk* − 1) yang paling dekat dengan bangun lingkaran. Selanjutnya, persamaan kurva lingkaran tersebut bisa dilihat seperti di bawah ini :

$$f(x, y) = x^2 + y^2 - r^2$$

Sehingga,

$$f\ (x, y) \begin{cases} < 0, & \text{jika } (x, y) \text{ didalam lingkaran} \\ = 0, & \text{jika } (x, y) \text{ terletak pada lingkaran} \\ > 0, & \text{jika } (x, y) \text{ terletak diluar lingkaran} \end{cases}$$

Dalam menentukan titik antara (*xk*+1, *yk*) dan (*xk*+1, *yk* − 1) dimana kita anggap paling dekat dengan lingkaran, terdapat beberapa ketentuan yang kita pakai dengan menggunakan letak kordinat tengah (*xk*+1, *yk* − ½) yaitu :

$$p_k = f\ (x_k + 1, y_k - \tfrac{1}{2}) = (x_k + 1)^2 + (y_k - \tfrac{1}{2})^2 - r^2$$

Apabila *p*k< 0 letak titik tengahnya terletak didalam dari lingkaran, artinya kordinat (*xk*+1, *yk*) yang kita pilih. Apabila *p*k> 0 letak titik tengahnya terletak diluar dari lingkaran, artinya kordinat (*xk*+1, *yk* − 1) yang kita pilih.

$$p_{k+1} = f\left(x_{k+1}+1, y_{k+1} - \tfrac{1}{2}\right) = [(x_k+1)+1]^2 + \left(y_{k+1} - \tfrac{1}{2}\right)^2 - r^2$$

Dan

$$p_{k+1} = p_k + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1$$

Titik *yk+1* yaitu *yk/yk*-1 bergantung pada tanda yang ada di *pk*. Untuk kasus seperti ini ketentuan awal dapat kita hitung menggunakan cara berikut :

$$p_0 = f(1, r - \tfrac{1}{2}) = 1 + (r - \tfrac{1}{2})^2 - r^2 = \tfrac{5}{4} - r$$

Apabilia nilai jari-jari lingkaran berupa bilangan bulat (integer), maka kita dapat melakukan pembulatan p0 seperti dibawah ini :

$$p_0 \square 1 \square r$$

Apabila nilai *pk* < 0 maka:

$$p_{k\square 1} \square p_k \square 2x_{k\square 1} \square 1$$

Apabila nilai *pk* > 0 artinya:

$$p_{k\square 1} \square p_k \square 2x_{k\square 1} \square 1 \square 2\, y_{k\square 1}$$

Sehingga kita memperoleh cara dalam membuat algoritma pembentukan lingkaran midpoint seperti dibawah ini :

1) Pilihlah *r* (jari-jari) kemudian letak titik pusat dari lingkaran (*xp*, *yp*), selanjutnya buatlah pengaturan sehingga posisi letak kordinat awal dengan nilai : (x_0, y_0) = (0 , *r*)
2) Hitunglah angka acuan dasar :

$$p_0 = \frac{5}{4} - r$$

Apabila *r* berupa bilangan pecahan

$p_0 = 1 - r$

Apabila *r* berupa bilangan bulat

3) Pada tiap-tiap letak posisi x_k, dapat kita mulai dengan k = 0 maka akan berlaku ketentuan seperti ini:
   a) jika $p_k < 0$, artinya kordinat berikutnya yaitu (x_k+1, y_k) kemudian $p_{k+1} = p_k + 2\,x_{k+1} + 1$
   b) jika bukan, artinya kordinat berikutnya yaitu (x_k+1, y_k – 1) kemudian $p_{k+1} = p_k + 2\,x_{k+1} + 1 - 2\,y_{k+1}$
4) Aturlah titik simetri di 7 (tujuh) oktan lainnya
5) Ubahlah posisi dari setiap pixel (*x, y*) di dalam garis bangun lingkaran yang berdasarkan kepada letak kordinat pusat (x_p, y_p) kemudian juga plot nilai kordinat : $x = x + x_p$, $y = y + y_p$
6) Ulang instruksi ke 3 dan juga instruksi ke 5 sampai didapatkan $x \geq y$

Contoh 1:

Coba buat kurva dari lingkaran yang memiliki titik pusat yaitu (4,6) dan jari-jari (*r*) sebesar 8. Lakukan perhitungan dengan didasarkan pada oktan kuadran pertama yaitu x = 0 hingga x =y, dengan kordinat awal di (x,r) = (0,8), diketahui jari-jari lingkaran adalah bilangan integer, oleh karena itu gunakan P0 + 1 - *r*.

Cara Menjawab :

Pengulangan pertama :

$K = 0$, $X_0 = 0$, $Y_0 = r = 8$, $P_0 = 1 - r = 1 - 8 = -7$

Sepertemuan $P_0 < 0$, oleh karena itu :

$X_1 = X_0 + 1 = 0 + 1 = 1$ kemudian $Y_1 = Y_0 = 8$,

Maka kordinat berikutnya : (1,8)

$P_1 = p_0 + 2\,x_1 + 1 = -7 + 2.(1) + 1 = -4$

Menggunakan lingkaran simetris 8 titik, Selanjutnya kita akan mendapatkan titik sebagaimana dibawah ini :

(1,8), (–1,8), (1, –8), (–1, –8), (8,1), (–8,1), (8, –1), (–8, –1)

Pindahkan letak tiap-tiap piksel (x,y) pada lingkaran yang memuliki titik pusat di (4,6) sehingga akan didapatkan letak titik sebagai berikut ini :

(5, 14), (3, 14), (5, –2), (3, –2), (12, 7), (–4, 7), (12, 5), (–4, 5)

Pengulangan kedua :

$K = 1$, $X_1 = 1$, $Y_1 = 8$, $P_1 = -4$

Sepertemuan $P_1 < 0$, oleh karena itu :

$X_2 = X_1 + 1 = 1 + 1 = 2$ kemudian $Y_2 = Y_1 = 8$,

Maka kordinatnya berikutnya : (2,8)

$P_2 = p_1 + 2\,x_2 + 1 = -4 + 2.(2) + 1 = 1$

Menggunakan lingkaran simetris 8 titik, Selanjutnya kita akan mendapatkan titik sebagaimana dibawah ini :

(2,8), (–2,8), (2, –8), (–2, –8), (8,2), (–8,2), (8, –2), (–8, –2)

Pindahkan letak tiap-tiap piksel (x,y) pada lingkaran yang memuliki titik pusat di (4,6) sehingga akan didapatkan letak titik sebagai berikut ini :

(6, 14), (2, 14), (6, –2), (2, –2), (12, 8), (–4, 8), (12, 4), (–4, 4)

Pengulangan ketiga :

$K = 2$, $X_2 = 2$, $Y_2 = 8$, $P_2 = 1$

Sepertemuan $P_2 > 0$, oleh karena itu :

$X_3 = X_2 + 1 = 2 + 1 = 3$ kemudian $Y_3 = Y_2 - 1 = 8 - 1 = 7$

Maka kordinat berikutnya : (3,7)

$P_3 = p_2 + 2\,x_3 + 1 - 2\,y_3 = 1 + 2.(3) + 1 - 2.(7) = -6$

Menggunakan lingkaran simetris 8 titik, Selanjutnya kita akan mendapatkan titik sebagaimana dibawah ini:

(3,7), (–3,7), (3, –7), (–3, –7), (7,3), (–7,3), (7, –3), (–7, –3)

Pindahkan letak tiap-tiap piksel (x,y) pada lingkaran yang memuliki titik pusat di (4,6) sehingga akan didapatkan letak titik sebagai berikut ini :

(7, 13), (1, 13), (7, –1), (1, –1), (11, 9), (–3, 9), (11, 3), (–3, 3)

Pengulangan keempat :

$K = 3$, $X_3 = 3$, $Y_3 = 7$, $P_3 = -6$

Sepertemuan $P_3 < 0$, oleh karena itu :

$X_4 = X_3 + 1 = 3 + 1 = 4$ kemudian $Y_4 = Y_3 = 7$,

Maka kordinat berikutnya : (4,7)

$P_4 = p_3 + 2\,x_4 + 1 = -6 + 2.(4) + 1 = 3$

Menggunakan lingkaran simetris 8 titik, Selanjutnya kita akan mendapatkan titik sebagaimana dibawah ini:

(4,7), (–4,7), (4, –7), (–4, –7), (7,4), (–7,4), (7, –4), (–7, –4)

Pindahkan letak tiap-tiap piksel (x,y) pada lingkaran yang memuliki titik pusat di (4,6) sehingga akan didapatkan letak titik sebagai berikut ini :

(8, 13), (0, 13), (8, –1), (0, –1), (11, 10), (–3, 10), (11, 2), (–3, 2)

engulangan kelima :

$K = 4$, $X_4 = 4$, $Y_4 = 7$, $P_4 = 3$

Sepertemuan $P_4 > 0$, oleh karena itu :

$X_5 = X_4 + 1 = 4 + 1 = 5$ kemudian $Y_5 = Y_4 - 1 = 7 - 1 = 6$,

Maka kordinat berikutnya: (5,6) $P_5 = p_4 + 2\,x_4 + 1 - 2\,y_4 = 3 + 2.(5) + 1 - 2.(6) = 2$

Menggunakan lingkaran simetris 8 titik, Selanjutnya kita akan mendapatkan titik sebagaimana dibawah ini:

(5,6), (–5,6), (5, –6), (–5, –6), (6,5), (–6,5), (6, –5), (–6, –5)

Pindahkan letak tiap-tiap piksel (x,y) pada lingkaran yang memuliki titik pusat di (4,6) sehingga akan didapatkan letak titik sebagai berikut ini :

(9, 12), (–1, 12), (9, 0), (–1, 0), (10, 11), (–2, 11), (10, 1), (–2, 1)

engulangan keenam :

$K = 5$, $X_5 = 5$, $Y_5 = 6$, $P_5 = 2$

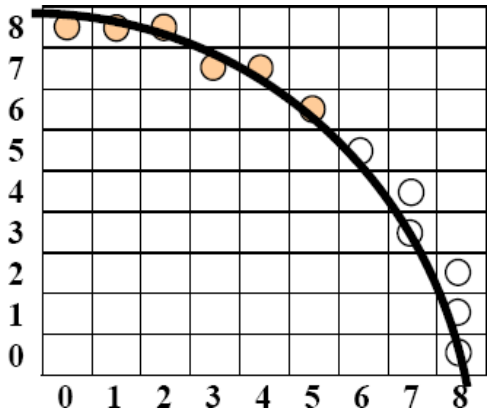
Sepertemuan $P_5 > 0$, oleh karena itu :

$X_6 = X_5 + 1 = 5 + 1 = 6$ kemudian $Y_6 = Y_5 - 1 = 6 - 1 = 5$,

Maka kordinat berikutnya: (6,5)

Pengulangan tidak lagi perlu dilakukan dikarenakan X > Y.

Jika dilakukan penggambaran, maka akan didapatkan hasil oktan pertama sebagaimana dapat dilihat pada gambar 4.4



Gambar 4. 4 Ilustrasi letak piksel dengan pembentukan lingkaran yang memiliki titik pusat yaitu (0,0) dengan r sebesar 8

2. Algoritma Pembentukan Elips

Bentuk elips adalah salah satu bentuk modifikasi dari bangun dasar lingkaran. Hal ini dilakukan dengan cara menambahkan beberapa minor dan mayor axis dalam tata cara pembentukan lingkaran. Bentuk ellips dibuat dengan satu bagian titik dan juga tetap memperhatikan gap diatara kedua titik atau biasa disebut foci. Jika gap kedua foci dari sembarang letak kordinatp (x,y) ditandai dengan tanda d1 dan juga d2, maka kita akan mendapatkan persamaan bentuk elips yaitu :

d1+d2=konstan

Hal ini untuk memberikan gambaran jarak antara d1 dan d2 dan kordinat setiap titik $F_{1(x1,y1)}$ kemudian $F_{2(X2,Y2)}$

((x-x1)2+(y-y1)2)2+((x-x2)2+(y-y2)2=konstan

Yaitu ketika posisi minor dan juga mayor axis bentuk ellips berada di posisi paralel antara sumbu x dengan sumbu y di dalam contoh seperti ini. Selain itu, parameter lainnya seperti rx dikenal dengan sebutan semi mayor axis dan ry

dikenal dengan sebutan semi minor axis. Maka akan diperoleh persamaan bentuk ellips menggunakan acuan dasar rx dan juga ry sebagai berikut :

$$((x-rc)/rx)2 + ((y-yc)/ry)2=1$$

Salah satu algoritma pembentukan ellips yang umum digunakan adalah algoritma elips midpoint. Algoritma ini memiliki pendekatan yang tidak berbeda jauh dengan algoritma pembentukan lingkaran. Menggunakan parameter dengan elips yang berada pada posisi yang umum, terdapat rx, ry, dan (xc,yc) di dalamnya. Jika elips akan digambarkan dengan letak pada umumnya, maka langkah yang bisa dipakai adalah mengubah ellips kearah turun (diputar) dari kordinat pusatnya, hal ini dilakukan supaya didapatkan kembali minor dan mayor axisnya.

Teknik ini dioperasikan di kuadran pertama yang terbagi ke dalam dua langkah, pertama yaitu menurut kemiringan dari bentuk elips rx<ry, apabila slope nilainya kurang dari 1 (satu) maka dilakukan penambahan unit step searah dengan sumbu x. Sedangkan apabila slope nilainya lebih besar dari 1 (satu) maka penambahan dilakukan searah dengan sumbu y.

Kedua bagian ini bisa diaplikasikan dalam berbagai macam teknik. Contoh pertama kita mulai dari letak posisiv(0,ry) dan step yang arahnya sama dengan jarum jam yang sejajar dengan jalur ellips dalam kuadran kesatu. Perubahan pada unit step di dalam x dilakukan ketika nilai kemiringannya memiliki angka diatas 1 (satu).

Teknik lainnya adalah dengan memulainya dari posisi (rx,0) dengan seleksi titik yang arahnya tidak searah dengan jarum jam. Pengubahan unit step y menjadi unit step x dilakukan ketika nilai kemiringan (*slope*) memiliki nilai diatas -1. Menggunakan prosesor paralel, maka kita dapat melakukan penghitungan posisi pixel dengan dua bagian secara bersamaan.

Teknik pembentuk elips dengan menggunakan algoritma midpoint adalah seperti di bawah ini :

1. Tentukan dimana posisi rx,ry kemudian juga titik inti dari ellips (xc,yc) hingga didapatkan

   $$(xo,yo)=(0,ry)$$

2. Hitunglah angka-angka pada acuan dasar : $P10=ry^2- rx^2 ry+1/4 rx^2$

3. Tentukan angka untuk awalan k=0, pada tiap-tiap letak posisi x_k maka digunakan ketentuan seperti dibawah ini :

   a. Jika nilai p1k< 0 artinya titik berikutnya yaitu (xk+1, yk)

$$p1_{k+1}=p1_k+2r_y^2x_{k+1}+r_y^2$$

   b. Jika tidak, artinya titik berikutnya yaitu (x_k+1,y_k-1) dan

$$p1_{k+1}=p1_k+2r_y^2x_{k+1}-2r_x\ y_{k+1}+r_y^2$$

Kemudian

$$2r_y^2x_{k+1}=2r_y^2x_k+2r_y^2$$

Dan

$$2r_x\ y_{k+1}=2r_x^2\ y_k+2r_x^2$$

Terakhir

$$2r_y^2x >= 2r_x^2\ y$$

4. Tentukan angka-angka dari acuan dasar di bagian selanjutbya dengan memakai titik akhir (x0,y0) dan sudah di lakukan perhitungan di bagian sebelumnya, seperti dapat dilihat dibawah ini :

$$P^2_{k+1=}2r_y^2\,(x_o+1/2)^2+2r_x^2\,(y_o-1)^2-\ r_x^2\ r_y^2$$

5. seluruh posisi y_k di bagian kedua kita mulai menggunakan k=0

   a. Jika p2k> 0, artinya kordinat berikutnya yaitu (xk, yk-1)

$$p2_{k+1}=p2_k+2r_x^2y_{k+1}+r_x^2$$

   b. Jika bukan, artinya titik berikutnya yaitu (x_k+1,y_k-1) dan

$$p2_{k+1}=pk+2r_y^2x_{k+1}-2r_x\ y_{k+1}+r_y^2$$

6. Carilah letak titik simetris di tiga bagian kuadran yang lain.
7. Ubahlah tiap-tiap posisi (x,y) di garis yang melingkar pada bentuk elips menggunakan titik pusat (xc,yc) kemudian carilah nilai kordinat

   X = x+ xc, y = y+yc

8. Terus ualng instruksi kesatu seperti diatas sampai didapatkan nilai $2r_y^2x$ > = $2r_x^2y$

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 4	1. Coba gambarkan lingkaran denga kordinat pusat yaitu (0,0) kemudian angka jari-jari (*r*)=6, hitunglah dengan didasarkan pada oktan kuadran pertama, dengan x = 0 hingga y = r, dan kordinat dari letak awalan yang diawali di titik (x,r) = (0,6).

D. REFERENSI

PERTEMUAN 5

ALGORITMA PEMBUATAN KURVA

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan algoritma pembautan.

Pada pertemuan ini akan dijelaskan mengenai :

1. Kurva
2. Algoritma Bezier

B. URAIAN MATERI

1. Kurva

Dalam matematika sebuah kurva adalah suatu objek geometri yang merupakan satu dimensi dan kontinyu.banyak kurva khusus telah dipelajari dalam geometri.

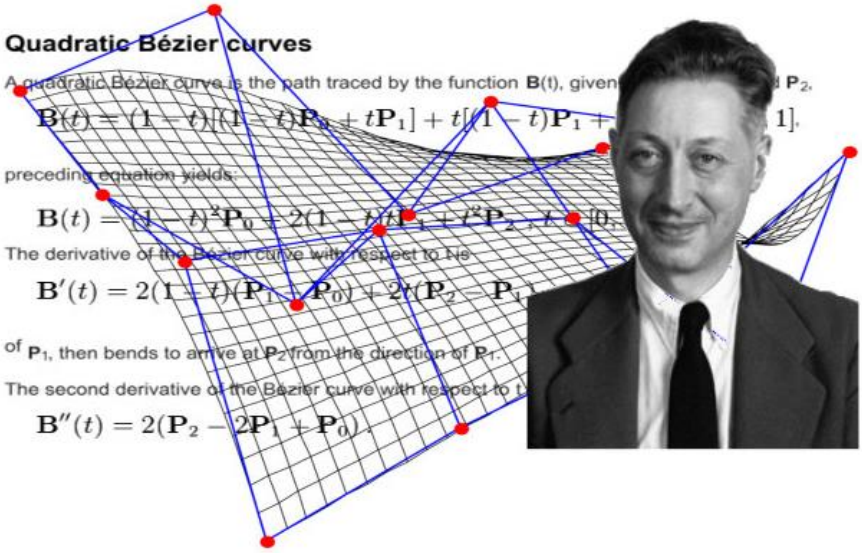
Geometri sering dikenal dengan istilah kurva (*curve*) dan permukaan (*surface*). Kurva merupakan suatu titik yang dibentuk dengan garis sehingga membentuk suatu lengkungan. Jenis-jenis kurva antara lain :

a. **Kurva linier** yaitu kurva yang dibentuk dari suatu garis antara 2 buah titik yang saling berhubungan.
b. **Kurva kubik** yaitu kurva yang memiliki persamaan ax3+bx2+cx+d di mana a,b,c,d adalah konstanta
c. **Kurva Bezier** yaitu kurva yang proses pembentukannya dari kurva linier,kurva kubik,kuadarat, Permukaan (surface) merupakan struktur matematis yang terbentuk atas himpunan kurva.

Kali ini kita akan mempelajari bagaimana membuat kurva menggunakan algoritma Bezier. Algoritma membuat kurva bezier diusulkan oleh seorang ahli mesin dari perancis yang bekerja di perusahaan Renault dan digunakan untuk merancang badan mobil.

2. Algoritma Bezier

Pembuatan kurva yang hendak dibahas dengan memakai algoritma yang diusulkan oleh Bezier.



Gambar 5. 1 Bezier

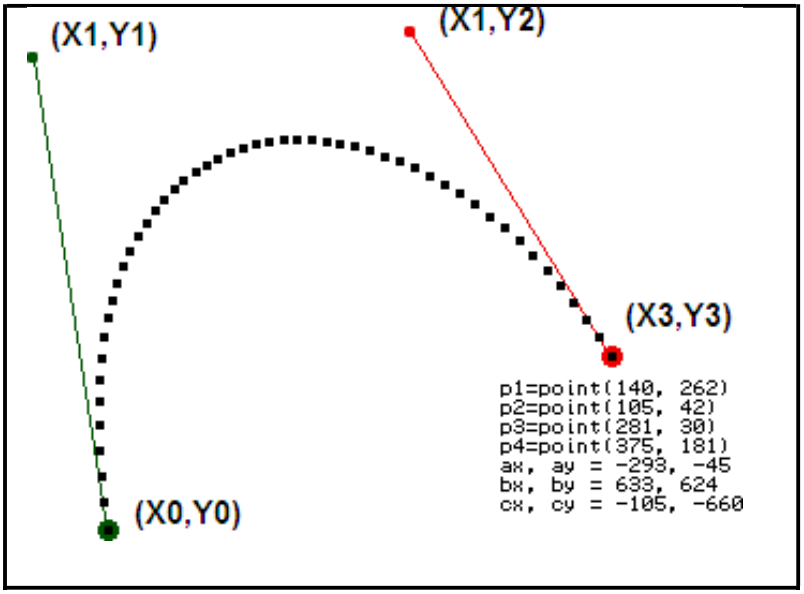
Pierre Bezier seorang ahli mesin perancis yang bekerja diperusahaan renult. Bezier lahir pada bersamaan pada 1 September 1910 dan meninggal pada tgl 25 Novmber 1999. Bezier memperoleh gelar dalam bidang mekanikal dari Ecole Nationale Superieure d' Seni et Metiers tahun 1930. Gelar kedua dibidang elektro pada tahun 1931 di Ecole Superieure d' Electricite, dan doktor pada tahun 1977 bidang matematik dari Universitas Paris. Ia bekerja buat renult dari 1933- 1975, di mana dia tingkatkan UNISURF USD CAM sistem. Dari tahun 1958- 1979 dia profesor penciptaan di tata cara konservatori nasional et des seni metiers. Pada tahun 1985 ia telah diakui oleh ACM SIGGRAPH dengan Steven A Coons atas kontribusi buat komputer grafis dan tata cara interaktif. Penafsiran Kurva Bezier.

a. Pengertian Kurva Bezier

Kurva Bezier diterapkan di bidang grafika computer buat menghasilkan kurva yang halus pada berbagai skala. Kurva Bezier diberinama sesuai dengan penemunya yakni Dokter. Pierre Bezier.

Dokter. Pierre Bezier merupakan seseorang engineer pada industri mobil Renault, persamaan Bezier sendiri dibesarkan pada tahun 1960- an serta digunakan buat desain body mobil.

Kurva Bezier awal adalah fungsi parametric yang terdiri dari 4 titik, titik awal, titik akhir dan 2 (dua) buah titik control.



Gambar 5. 2 Fungsi kurva Bezier dengan 4 titik

Keterangan :

(x_0,y_0) adalah titik awal .

(x_3,y_3) adalah titik akhir.

Titik (x_1,y_1) dan (x_2,y_2) adalah titik kontrol

Bentuk umum dari Persamaan Bezier adalah fungsi yang menggambarkan tiap titik pada kurva sebagai fungsi waktu.

1) Persamaan Kurva Bezier

Persamaan kurva Bezier disusun berdasarkan persamaan polinomial Bernstein berikut :

$$B_n(t) = \sum_{i=0}^{n} \binom{n}{i} t^i \cdot (1-t)^{n-i}$$

Gambar 5. 3 Persamaan Polinomial Bernstein

Pada kurva Bezier t dibatasi pada interval 0..1. Persamaan sebuah kurva Bezier dengan n titik control {P_1, P_2, P_3…P_n} adalah sebagai berikut :

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} t^i \cdot (1-t)^{n-i} \cdot P_i$$

Gambar 5. 4 Persamaan kurva Bezier dengan n titik control

Kurva Bezier dimulai pada titik P_0 dan berakhir di titik P_n , tetapi belum tentu melewati titik-titik kontrolnya.

Untuk menggambarkan titik-titik pada kurva Bezier maka dilakukan perhitungan titik dari P0 sampai dengan Pn dengan t dari 0 sampai dengan 1, dengan Δt yang bervariasi, contoh untuk Δt = 0,1 maka jumlah titik yang harus digambarkan dapat dilihat pada tabel berikut :

Table 5. 1 Tabel titik kurva Bezier untuk Δt=0.1

T	(x,y)
0	Titik pertama
0.1	Titik kedua
0.2	Titik ketiga
0.3	Titik keempat

T	(x,y)
0.4	Titik kelima
0.5	Titik keenam
0.6	Titik ketujuh
0.7	Titik kedelapan
0.8	Titik kesembilan
0.9	Titik kesepuluh
1	Titik kesebelas

Jika Δt = 0,01, maka titik yang harus digambarkan sebanyak 101, dari t mulai 0, 0.01, 0.02, dst sampai t =1, semakin kecil Δt maka akan semakin banyak titik yang harus dihitung dan akan semakin halus kurva Bezier yang dihasilkan.

Pertanyaan :

Berapa jumlah titik yang harus digambarkan jika $\Delta t = n$

Jawab :

Jumlah titik yang harus digambarkan adalah : 1/n + 1 (dibulatkan)

Perhitungan koordinat titik (x,y) dapat dihitung menggunakan persamaan pada gambar 5.4, selain dengan cara tersebut dapat dibantu menggunakan segitiga pascal seperti terlihat pada contoh.

Contoh 1:

Diketahui 3 buah titik kontrol dengan koordinat C1(1,2), C2(7,10), C3(15,4) , dengan menggunakan Δt =0.02 maka tentukanlah:

a) Berapa titik yang digunakan untuk membangun kurva bezier?
b) Berapa nilai titik pada kurva pada saat t=0.8?

Jawab :

a) Dengan kenaikan sebanyak 0.02 maka jumlah titik yang diperlukan antara 0 dan 1 adalah

$$\frac{1}{t}+1=\frac{1}{0.02}+1=51$$

b) Sebenarnya titik yang harus digambar adalah 51 titik, tetapi untuk soal ini yang dicari hanya koordinat titik pada saat t = 0.8

Table 5. 2 Tabel titik kurva Bezier untuk Δt=0.01

T	(x,y)
0	
0.02	
0.04	
0.06	
..	
..	
0.8	
..	
..	
1	

Bisa dicari dengan memasukkan pada persamaan pada gambar atau atau bisa dengan perhitungan berikut.

Karena terdiri dari 3 titik kontrol maka persamaan menjadi : $(x+y)^{3-1}$ =

$(x+y)^2 = x^2 + 2xy + y^2 = 0$ Ganti x = (1-t) dan y = t Maka persamaan tersebut menjadi :

$L(t) = (1-t)^2 + 2(1-t)t + t^2$

Koordinat titik untuk t = 0.8

<u>Sumbu x</u>

$X(0,8) = (1-t)^2.X_1 + 2(1-t)t.X_2 + t^2.X_3$

Keterangan :

X_1,X_2 dan X_3 diambil dari titik control yaitu 1, 7 dan 15 $X(0.8) = (1-0.8)^2.1 + 2(1-0.8).(0.8).7 + (0.8)^2.1$

$= 0.04 + 2.24 + 9.6 = 11.88 \sim 12$

<u>Sumbu Y</u>

$Y(0.8) = (1-t)^2.Y_1 + 2(1-t)t.Y_2 + t^2.Y_3$

Keterangan :

Y_1, Y_2 dan Y_3 diambil dari titik control yaitu 2, 10 dan 4 $Y(0.8) = (1-0.8)^2.2 + 2(1-0.8).(0.8).10 + (0.8)^2.4$

$= 0.08 + 3.2 + 2.56 = 5.84 \sim 6$

Maka koordinat titik pada saat t=0.8 adalah (12,6)

Contoh 2:

Diketahui 4 buah titik kontrol dengan koordinat C1(0,1), C2(1,2), C3(2,2), C4(3,1) dengan menggunakan kenaikan t=0.02 maka tentukanlah berapa nilai titik pada kurva pada saat t=0.8?

<u>Jawab :</u>

Karena terdiri dari 4 titik kontrol maka persamaan menjadi :

$(x+y)^{4-1} = (x+y)^3$

$= x^3 + 3x^2y + 3xy^2 + y^3 = 0$ Ganti x = (1-t) dan y = t

Maka persamaan tersebut menjadi :

$L(t) = (1-t)^3 + 3(1-t)^2t + 3(1-t)t^2 + t^3$

Koordinat titik untuk t = 0.8

<u>Sumbu x</u>

$X(0.8) = (1-t)^3.X_1 + 3(1-t)^2t.X_2 + 3(1-t)t^2.X_3 + t^3.X_4$

Keterangan :

X_1,X_2, X_3 dan X_4 diambil dari titik control yaitu 0, 1, 2 dan 3 $X(0.8) = (1-0.8)^3.0+3(1-0.8)^2.(0.8).1+3(1-0.8).(0.8)^2.2+ (0.8)^3.3$

= 0 + 0.096 + 0.768 + 1.536 = 2.4 ~ 2

<u>Sumbu Y</u>

$Y(0.8) = (1-t)^3.Y_1 + 3(1-t)^2t.Y_2 + 3(1-t)t^2.Y_3 + t^3.Y_4$

Keterangan :

Y_1,Y_2, Y_3 dan Y_4 diambil dari titik control yaitu 1, 2, 2 dan 1 $Y(0.8) = (1-0.8)^3.1+3(1-0.8)^2.(0.8).2+3(1-0.8).(0.8)^2.2+ (0.8)^3.1$

= 0.008+0.192+0.768+0.512 = 1.48 ~ 1

Maka koordinat titik pada saat t=0.8 adalah (2,1)

Panduan untuk persamaan polinomial

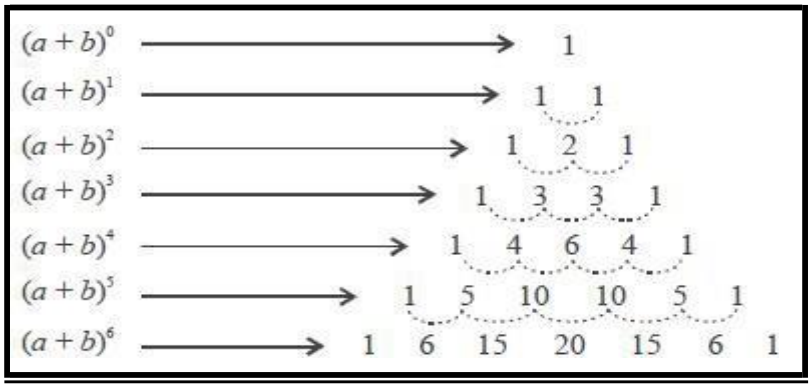
Untuk membantu membentuk persamaan polinomial pada kurva bezier berikut adalah panduannya, kurva bezier dengan n titik kontrol maka

persamaannya adalah :

$$\boxed{(x+y)^{n-1}}$$

Gambar 5. 5 Persamaan polinomial untuk kurva bezier dengan n titik control

Pembentukan persamaan polinomial dapat dibantu dengan segitiga pascal seperti terlihat pada gambar 5.6



Gambar 5. 6 Segitiga pascal dan persamaan polynomial

Angka pada segitiga pascal tersebut mewakili koefisien dari persamaan polynomial, Lihat contoh 2 untuk kurva Bezier dengan 4 titik control maka persamaannya adalah $(x+y)^3$ atau lihat pada gambar adalah $(a+b)^3$ Maka koefisien persamaan polynomial adalah 1 3 3 1 (lihat gambar 5.5), dapat dituliskan sebagai berikut :

$$1X^3Y^0 + 3X^2Y^1 + 3X^1Y^2 + 1X^0Y^3$$

Atau

$$X^3 + 3X^2Y + 3XY^2 + Y^3$$

Pangkat pada X dimulai dengan pangkat pada persamaan yaitu 3, kemudian dikurangi 1, sebaliknya pangkat pada Y dimulai dengan 0 dan bertambah 1

Tabel 5.1 tabel titik kurva Bezier untuk $\Delta t=0.1$

T	(x,y)
0	Titik pertama
0.1	Titik kedua
0.2	Titik ketiga
0.3	Titik keempat

0.4	Titik kelima
0.5	Titik keenam
0.6	Titik ketujuh
0.7	Titik kedelapan
0.8	Titik kesembilan
0.9	Titik kesepuluh
1	Titik kesebelas

Jika Δt = 0,01, maka titik yang harus digambarkan sebanyak 101, dari t mulai 0, 0.01, 0.02, dst sampai t =1, semakin kecil Δt maka akan semakin banyak titik yang harus dihitung dan akan semakin halus kurva Bezier yang dihasilkan.

Pertanyaan :

Berapa jumlah titik yang harus digambarkan jika $\Delta t = n$

Jawab :

Jumlah titik yang harus digambarkan adalah : 1/n + 1 (dibulatkan)

Perhitungan koordinat titik (x,y) dapat dihitung menggunakan persamaan pada gambar 5.4, selain dengan cara tersebut dapat dibantu menggunakan segitiga pascal seperti terlihat pada contoh.

Contoh 1 :

Diketahui 3 buah titik kontrol dengan koordinat C1(1,2), C2(7,10), C3(15,4) , dengan menggunakan Δt =0.02 maka tentukanlah:

1. Berapa titik yang digunakan untuk membangun kurva bezier?
2. Berapa nilai titik pada kurva pada saat t=0.8?

Jawab :

1. Dengan kenaikan sebanyak 0.02 maka jumlah titik yang diperlukan antara 0 dan 1 adalah

$$\frac{1}{t}+1=\frac{1}{0.02}+1=51$$

2. Sebenarnya titik yang harus digambar adalah 51 titik, tetapi untuk soal ini yang dicari hanya koordinat titik pada saat t = 0.8

Tabel 5.2 tabel titik kurva Bezier untuk Δt=0.01

T	(x,y)
0	
0.02	
0.04	
0.06	
..	
..	
0.8	
..	
..	
1	

Bisa dicari dengan memasukkan pada persamaan pada gambar atau atau bisa dengan perhitungan berikut. Karena terdiri dari 3 titik kontrol maka persamaan menjadi : $(x+y)^{3-1} = (x+y)^2 = x^2 + 2xy + y^2 = 0$ Ganti x = (1-t) dan y = t

Maka persamaan tersebut menjadi :

$L(t) = (1-t)^2 + 2(1-t)t + t^2$

Koordinat titik untuk t = 0.8

<u>Sumbu x</u>

$X(0,8) = (1-t)^2.X_1 + 2(1-t)t.X_2 + t^2.X_3$

Keterangan :

X_1,X_2 dan X_3 diambil dari titik control yaitu 1, 7 dan 15 $X(0.8) = (1-0.8)^2.1 + 2(1-0.8).(0.8).7 + (0.8)^2.15$

$= 0.04 + 2.24 + 9.6 = 11.88 \sim 12$

<u>Sumbu Y</u>

$Y(0.8) = (1-t)^2.Y_1 + 2(1-t)t.Y_2 + t^2.Y_3$

Keterangan :

Y_1, Y_2 dan Y_3 diambil dari titik control yaitu 2, 10 dan 4 $Y(0.8) = (1-0.8)^2.2 + 2(1-0.8).(0.8).10 + (0.8)^2.4$

$= 0.08 + 3.2 + 2.56 = 5.84 \sim 6$

Maka koordinat titik pada saat t=0.8 adalah (12,6)

Contoh 2 :

Diketahui 4 buah titik kontrol dengan koordinat C1(0,1), C2(1,2), C3(2,2), C4(3,1) dengan menggunakan kenaikan t=0.02 maka tentukanlah berapa nilai titik pada kurva pada saat t=0.8?

<u>Jawab :</u>

Karena terdiri dari 4 titik kontrol maka persamaan menjadi :

$(x+y)^{4-1} = (x+y)^3$

$= x^3 + 3x^2y + 3xy^2 + y^3 = 0$ Ganti x = (1-t) dan y = t

Maka persamaan tersebut menjadi :

$L(t) = (1-t)^3 + 3(1-t)^2t + 3(1-t)t^2 + t^3$

Koordinat titik untuk t = 0.8

Sumbu x

$X(0.8) = (1-t)^3.X_1 + 3(1-t)^2t.X_2 + 3(1-t)t^2.X_3 + t^3.X_4$

Keterangan :

X_1,X_2, X_3 dan X_4 diambil dari titik control yaitu 0, 1, 2 dan 3 $X(0.8) = (1-0.8)^3.0+3(1-0.8)^2.(0.8).1+3(1-0.8).(0.8)^2.2+ (0.8)^3.3$

$= 0 + 0.096 + 0.768 + 1.536 = 2.4 \sim 2$

Sumbu Y

$Y(0.8) = (1-t)^3.Y_1 + 3(1-t)^2t.Y_2 + 3(1-t)t^2.Y_3 + t^3.Y_4$

Keterangan :

Y_1,Y_2, Y_3 dan Y_4 diambil dari titik control yaitu 1, 2, 2 dan 1 $Y(0.8) = (1-0.8)^3.1+3(1-0.8)^2.(0.8).2+3(1-0.8).(0.8)^2.2+ (0.8)^3.1$

$= 0.008+0.192+0.768+0.512 = 1.48 \sim 1$

Maka koordinat titik pada saat t=0.8 adalah (2,1)

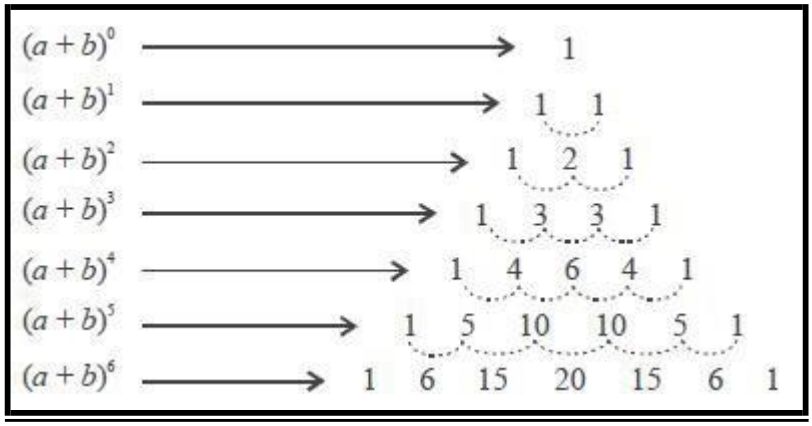
Panduan untuk persamaan polinomial

Untuk membantu membentuk persamaan polinomial pada kurva bezier berikut adalah panduannya, kurva bezier dengan n titik kontrol maka persamaannya adalah :

$$\boxed{(x+y)^{n-1}}$$

Gambar 5. 7 Persamaan polinomial untuk kurva bezier dengan n titik kontrol

Pembentukan persamaan polinomial dapat dibantu dengan segitiga pascal seperti terlihat pada gambar 5.6



Gambar 5. 8 Segitiga pascal dan persamaan polynomial

Angka pada segitiga pascal tersebut mewakili koefisien dari persamaan polynomial, Lihat contoh 2 untuk kurva Bezier dengan 4 titik control maka persamaannya adalah $(x+y)^3$ atau lihat pada gambar adalah $(a+b)^3$ Maka koefisien persamaan polynomial adalah 1 3 3 1 (lihat gambar 5.5), dapat dituliskan sebagai berikut :

$$1X^3Y^0 + 3X^2Y^1 + 3X^1Y^2 + 1X^0Y^3$$

Atau

$$X^3 + 3X^2Y + 3XY^2 + Y^3$$

Pangkat pada X dimulai dengan pangkat pada persamaan yaitu 3, kemudian dikurangi 1, sebaliknya pangkat pada Y dimulai dengan 0 dan bertambah 1

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 5	1. Bagaimana persamaan polinomial untuk kurva bezier dengan 6 titik, 7 titik dan 8 titik kontrol 2. Diketahui 6 buah titik kontrol dengan koordinat C0(-45,-15), C1(-40,- 12), C2(-32,-2), C3(-24,0), C4(10,12), C5(24,20) (dengan menggunakan Δt=0.065 maka tentukanlah: a. berapa jumlah titik yang harus digambar untuk membentuk kurva bezier b. berapa nilai titik pada saat t =0.55

D. REFERENSI

PERTEMUAAN 6

ALGORITMA PEWARNAAN BIDANG

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu algoritma pewarnaan bidang.

Pada pertemuan ini akan dijelaskan mengenai :

1. Pengertian Pewarnaan Bidang
2. Algoritma Pewarnaan Bidang

B. URAIAN MATERI

1. Pengertian Pewarnaan Bidang

Dalam teori graf, pewarnaan graf merupakan permasalahan spesial pelabelan graf; merupakan penugasan label yang secara tradisional diucap "corak" ke elemen grafik yang tunduk pada batas tertentu. Dalam wujud yang sangat simpel, ini merupakan metode memberi warna simpul- simpul dari suatu grafik sehingga tidak terdapat 2 simpul yang bersebelahan mempunyai corak yang sama; ini diucap pewarnaan puncak. Demikian pula, corak tepi menetapkan corak ke tiap tepi sehingga tidak terdapat 2 tepi yang bersebelahan mempunyai corak yang sama, serta corak wajah dari bagan datar menetapkan corak ke tiap wajah ataupun kawasan sehingga tidak terdapat 2 wajah yang berbagi. batasan dengan corak yang sama. corak yang sama.

Pewarnaan simpul biasanya digunakan buat menghadirkan permasalahan pewarnaan grafis, sepertemuan permasalahan pewarnaan yang lain bisa diganti jadi contoh pewarnaan simpul. Bagaikan contoh, pewarnaan tepi dari suatu graf cumalah pewarnaan simpul dari graf garisnya, serta pewarnaan wajah dari suatu graf datar cumalah pewarnaan simpul dari gandanya. Tetapi, permasalahan pewarnaan tanpa simpul kerap diajukan serta dipelajari sebagaimana terdapatnya. Ini sebagian bertabiat pedagogis serta sebagian sepertemuan

sebagian permasalahan sangat baik dipelajari dalam wujud tanpa titik sudutnya, semacam dalam permasalahan pewarnaan tepi.

Syarat pemakaian corak berasal dari memberi warna negara- negara di peta, di mana tiap wajah diwarnai secara harfiah. Ini digeneralisasikan buat memberi warna wajah dari grafik yang disematkan di pesawat. Dengan dualitas bidang itu jadi memberi warna simpul, serta dengan metode ini digeneralisasikan ke seluruh grafik. Dalam representasi matematika serta pc, umumnya memakai bilangan bundar positif ataupun non- negatif awal bagaikan "corak". Secara universal, tiap himpunan sampai bisa digunakan bagaikan "himpunan corak". Watak dari permasalahan pewarnaan bergantung pada jumlah corak, namun bukan pada jumlah rupanya.

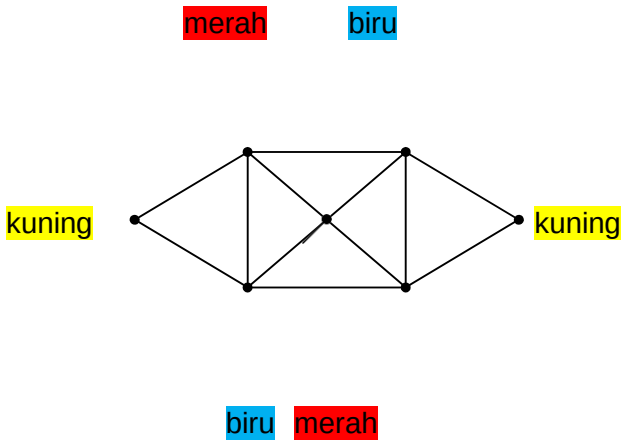
Pewarnaan grafis menikmati banyak aplikasi instan dan tantangan teoretis. Tidak hanya tipe permasalahan klasik, bermacam batas pula bisa diresmikan pada grafik, ataupun metode penetapan corak, ataupun apalagi corak itu sendiri. Dia apalagi menarik atensi warga universal dalam wujud misteri angka Sudoku yang terkenal. Pewarnaan grafis senantiasa jadi bidang riset yang sangat aktif.

2. Algoritma Perwanaan Bidang

Terdapat 3 berbagai pewarnaan graf, ialah pewarnaan simpul pewarnaan sisi, serta pewarnaan daerah (*region*). Yang hendak kita bahas merupakan pewarnaan simpul serta pewarnaan daerah (*region*).

Pewarnaan simpul merupakan berikan corak pada simpul-simpul sesuatu graf sedemikian sampai tidak terdapat 2 simpul bertetangga yang memiliki corak yang sama. Kita bisa membagikan sembarang corak pada simpul- simpul asalkan berbeda dengan simpul simpul tetangganya.

Dalam pewarnaan graf, kita tidak cuma hanya memberi warna simpul-simpul dengan corak yang berbeda dengan corak simpul tetangganya saja, tetapi kita pula menginginkan supaya jumlah corak yang digunakan sesedikit bisa jadi. Jumlah corak minimum yang bisa digunakan buat memberi warna simpul simpul diucap bilangan kromatik dari graf G, yang dinotasikan dengan G. Foto 1 memperlihatkan suatu graf, dengan G= 3.



Gambar 6. 1 Tiga warna cukup untuk mewarnai graf ini

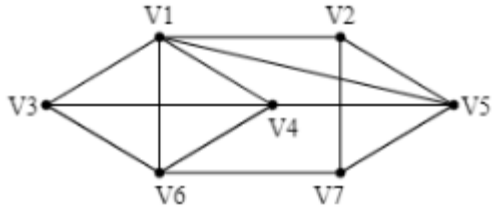
a. Algoritma Welch-Powell

Algoritma Welch- Powell merupakan sesuatu metode yang efektif buat memberi warna suatu graf G. tetapi algoritma ini cuma membagikan batasan atas untuk (G) Jadi algoritma ini tidak selalu memberikan jumlah warna minimum yang diperlukan Untuk mewarnai G. Menentukan (G) sesungguhnya sangat susah kecuali dalam kasus- kasus simpel semacam pada contoh- contoh yang hendak kita bahas dalam pertemuan ini.

Langkah- langkah dalam algoritma Welch- Powell:

a. Urutkan simpul- simpul dari G dalam urutan derajat yang menyusut. Urutan ini bisa jadi tidak unik sepertemuan sebagian simpul bisa jadi memiliki derajat yang sama.
b. Pakai satu corak tertentu buat memberi warna simpul awal. Secara berurut, tiap simpul dalam catatan yang tidak bertetangga dengan simpul tadinya diwarnai dengan corak ini.
c. Ulangi langkah 2 di atas buat simpul dengan urutan paling tinggi yang belum diwarnai.
d. Ulangi langkah 3 di atas hingga seluruh simpul dalam catatan terwarnai.

Pakai algoritma Welch- Powell buat memberi warna graf G yang ditunjukkan pada gambar 6.2 serta tentukan bilangan kromatiknya.



Gambar 6. 2 Pakai algoritma Welch- Powell buat memberi warna graf G

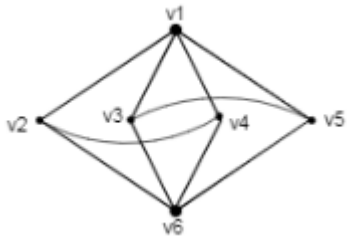
<u>Penyelesaian :</u>

Simpul	v1	v4	v5	v6	v2	v3	v7
Derajat	5	4	4	4	3	3	3
Warna	*a*	*b*	*c*	*c*	*b*	*d a*	

Jadi, paling tidak ada 4 warna diperlukan untuk mewarnai graf G, sehingga (*G*) = 4.

Contoh 2.

Permasalahan sama dengan contoh 1, untuk graf H yang ditunjukkan pada gambar 6.3 Penyelesaian :



Gambar 6. 3 Permasalahan isama idengan icontoh i1, iuntuk igraf iH

Simpul	v1	v6	v2	v3	v4	v5
Derajat	4	4	3	3	3	3
Warna	*a*	*a*	*b*	*b*	*c c*	

Jadi (*G*) = 3

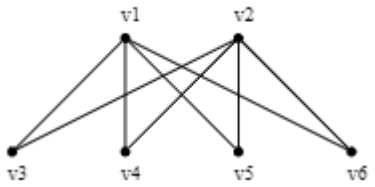
b. Pewarnaan pada Graf Bipartit

Sebuah graf bipartit adalah sebuah graf yang simpul-simpulnya dapat dibagi ke dalam dua himpunan bagian dimana simpul-simpul pada

masing-masing himpunan bagian bertetangga dengan semua simpul pada himpunan bagian lainnya dan bukan pada simpul-simpul dalam himpunan bagiannya sendiri. Karena tidak ada simpul-simpul yang bertetangga ke simpul-simpul yang bertetangga ke simpul lain dalam himpunan bagian yang sama, maka semua simpul dalam sebuah himpunan bagian dapat dipetakan ke dalam warna yang sama. Karena simpul-simpul pada dua himpunan bagian saling bertetangga, maka pada setiap himpunan bagian harus diwarnai dengan warna yang berbeda. Dengan demikian, dibutuhkan dua warna untuk mewarnai graf bipartit, sehingga bilangan kromatis pada graf bipartit adalah 2.

Contoh 3.

Diketahui sebuah graf bipartit $K_{2.4}$ seperti ditunjukkan pada gambar di bawah.



Gambar 6. 4 Graf bipartit K2.4

Dengan mengguanakan algoritma Welch-Powell, tentukan nilai kromatis dari graf di atas

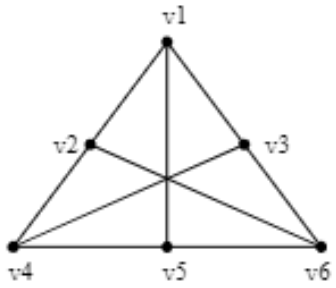
Simpul	v1	v2	v3	v4	v5	v6
Derajat	4	4	2	2	2	2
Warna	*a*	*a*	*B*	*b*	*b*	*B*

Jadi (G = 2, dan dapat dilihat bahwa dua himpunan bagian dalam graf bipartit tersebut adalah m = {v1, v2} dan n = {v3, v4, v5, v6}

Contoh 4

Graf G pada gambar 5 adalah graf bipartit. Petakan warna-warna ke simpul simpul dari G dengan menggunakan algoritma Welch Powell untuk

menunjukkan dua himpunan bagian dari simpul-simpul yang membangun G.



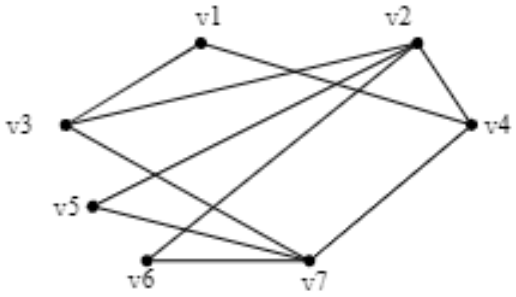
Gambar 6. 5 Graf bipartit

Simpul	v1	v2	v3	v4	v5	v6
Derajat	3	3	3	3	3	3
Warna	*a*	*b*	*b*	*a*	*b*	*a*

Jadi dua himpunan bagian yang membentuk *G* adalah *m* = {v1, v4, v6} *n* = {v2, v3, v5}

Contoh 5.

Permasalahan yang sama dengan contoh 4, pada graf G yang ditunjukkan pada gambar 6.6 di bawah ini.



Gambar 6. 6 Graf G

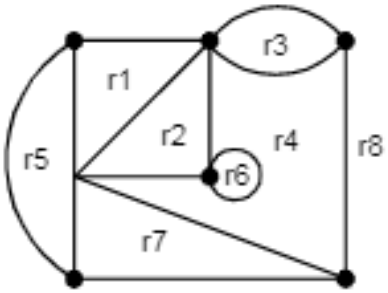
Simpul	v2	v7	v3	v4	v1	v5	v6
Derajat	4	4	3	3	2	2	2

Warna *a* *a* *B* *b* *A* *b* *b*

Jadi dua himpunan bagian yang membentuk *G* adalah
m = {v2, v7, v1} *n* = {v3, v4, v5, v6}

c. Pewarnaan Wilayah/Region pada Graf Bidang

Dua buah region dari sebuah graf bidang dikatakan bertetangga jika keduanya mempunyai sebuah sisi bersama.



Gambar 6. 7 Graf bidang

Dari sebuah graf bidang pada gambar 6.7, tentukan region dari graf tersebut yang bertetangga dengan region-region :

a. r7

b. r2

c. r6

<u>Penyelesaian :</u>

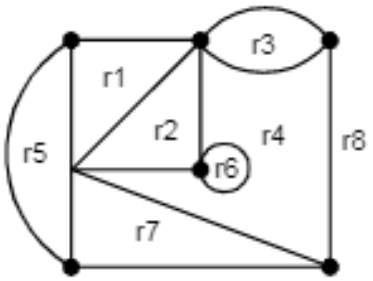
a. r4, r5, r8

b. r1, dan r4

c. r4

d. Pewarnaan Wilayah/Region pada Graf Bidang

Dua buah region dari sebuah graf bidang dikatakan bertetangga jika keduanya mempunyai sebuah sisi bersama.



Dari sebuah graf bidang pada gambar 6.7, tentukan region dari graf tersebut yang bertetangga dengan region-region :

a. r7
b. r2
c. r6

<u>Penyelesaian :</u>

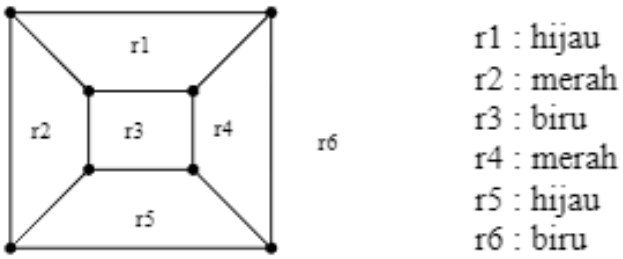
a. r4, r5, r8

b. r1, dan r4

c. r4

Pewarnaan Region (wilayah)

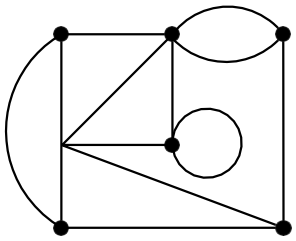
Pewarnaan region dari suatu graf planar (graf bidang) *G* adalah suatu pemetaan warna- warna ke region-region dari graf *G* sedemikian hingga region-region yang bertetangga mempunyai warna yang berbeda. Gambar 8 menunjukkan contoh permasalahan pewarnaan region.



Gambar 6. 8 Graf planar (graf bidang) G

Contoh 6.

Misal kita melakukan pewarnaan region dari graf pada gambar 7, yang hasilnya akan bisa dilihat seperti pada gambar 6.9 di bawah ini.



Gambar 6. 9 Pewarnaan region

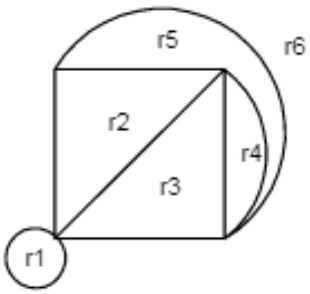
Pada gambar 6.9 bisa dilihat bahwa (*G*) = 3.

Graf Dual dari Graf Planar

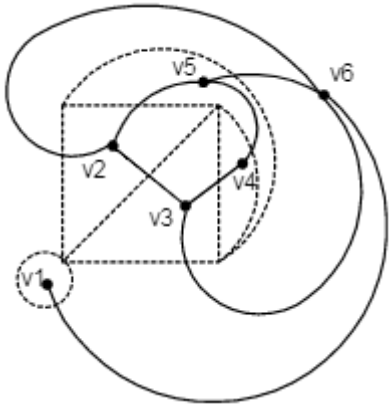
Dari suatu permasalahan pewarnaan region pada graf bidang, bisa kita bawa ke permasalahan pewarnaan simpul dengan membangun sebuah graf dual dari graf bidang tersebut.

Cara membentuk graf dual

Misal terdapat sebuah graf bidang M. Dalam setiap region dari M, pilih sebuah titik. Jika dua buah region mempunyai sebuah sisi bersama, maka titik-titik yang terkait dapat dihubungkan dengan sebuah garis melalui sisi bersama tersebut. iGaris-garis ini akan membentuk kurva. Kurva-kurva ini digambarkan sedemikian hingga agar tidak bersilangan. Dengan demikian kurva-kurva tersebut membentuk sebuah graf yang disebut sebagai graf dual dari M. Gambar 6.10 menunjukkan graf dual dari graf planar pada gambar 9.

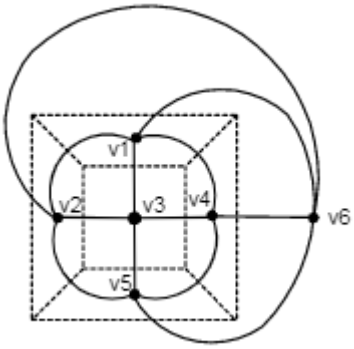


Gambar 6. 10 Graf bidang M



Gambar 6. 11 Pewarnaan simpul

Permasalahan pewarnaan region seperti yang ditunjukkan pada gambar 8 dapat kita bawa ke masalah pewarnaan isimpul, dengan ikita buat graf dual dari gambar 8 seperti ditunjukkan dalam gambar 6.11.



Gambar 6. 12 Algoritma Welch Powell

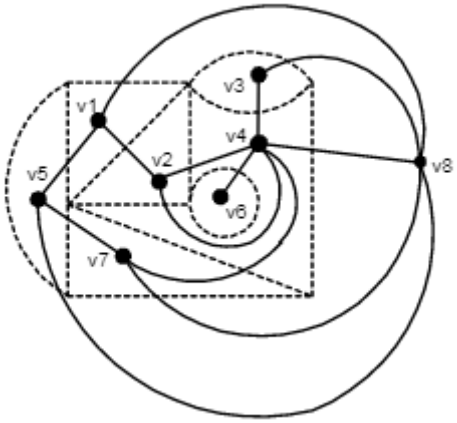
Dengan algoritma Welch Powell (permasalahan pewarnaan simpul),

Simpul	v1	v2	v3	v4	v5	v6
Derajat	4	4	4	4	4	4
Warna	*a*	*b*	*C*	*b*	*a*	*c*

(*G*) = 3. Hasil ini sama dengan hasil dari pewarnaan region pada gambar 8.

Contoh 7.

Permasalahan pada contoh 6 juga dapat kita bawa ke masalah pewarnaan simpul, dengan kita buat graf dual seperti ditunjukkan pada gambar 6.12.



Gambar 6. 13 Algoritma Welch Powell

Dengan algoritma Welch Powell,

Simpul	v4	v8	v1	v2	v5	v7	v3	v6
Derajat	6	5	3	3	3	3	2	1
Warna	*a*	*b*	*a*	*b*	*c*	*d*	*c*	*b*

(G) = 4. Hasil ini sama dengan hasil dari pewarnaan region pada contoh 6.13. Jika kita lihat pewarnaan region iyang kita lakukan sebelumnya pada subpertemuan 6.4, hasil ini memang berbeda. Ini adalah bukti bahwa algoritma welch Powell memang tidak selalu menghasilkan warna minimum (lihat kembali subpertemuan 6.2)

Contoh 8.

(Contoh aplikasi pewarnaan graf)

Ada 6 jenis zat kimia yang perlu disimpan di dalam gudang. Beberapa pasangan zat itu tidak dapat disimpan di dalam ruangan yang sama, karena campuran gasnya bersifat eksplosif (mudah meledak). Untuk zat

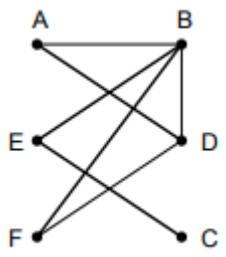
yang semacam itu, perlu dibangun ruang-ruang terpisah yang dilengkapi ventilasi dan penyedot udara keluar yang berlainan. Jika lebih banyak ruang yang dibutuhkan, berarti lebih banyak ongkos yang dikeluarkan. Karena itu perlu diketahui berapa banyak minimum ruangan yang diperlukan untuk dapat menyimpan semua zat kimia dengan aman. Berikut ini adalah daftar pasangan zat kimia yang tidak dapat disimpan dalam ruangan yang sama.

Zat Kimia	Tidak dapat disimpan bersama zat kimia
A	B, D
B	A, D, E, F
C	E
D	A, F, B
E	B, C
F	B, D

Gambar 6. 14 Gambarkan graf

Gambarkan igraf iyang imenyatakan ipersoalan idi iatas. iKemudian itentukan ijumlah iminimum iruangan iyang idibutuhkan iuntuk imenyimpan isemua izat ikimia idi iatas.

Graf yang merepresentasikan permasalahan di atas di tunjukkan pada gambar 6.14. Simpul-simpul ipada graf menyatakan masing-masing zat kimia. Sisi yang menghubungkan dua simpul menyatakan bahwa dua zat kimia yang terkait tidak dapat disimpan dalam ruangan yang sama.



Gambar 6. 15 Bilangan kromatik

Berdasarkan graf tersebut kita menyimpulkan, bahwa apabila terdapat dua simpul yang dihubungkan oleh sisi, maka kedua zat kimia tersebut tidak dapat tidak dapat disimpan dalam ruang yang sama, jadi dua simpul tersebut tidak boleh mempunyai warna yang sama. Permasalahan di atas, sama saja kita mencari bilangan kromatik dari graf yang ditunjukkan pada gambar 6.15.

Dengan algoritma Welch Powell,

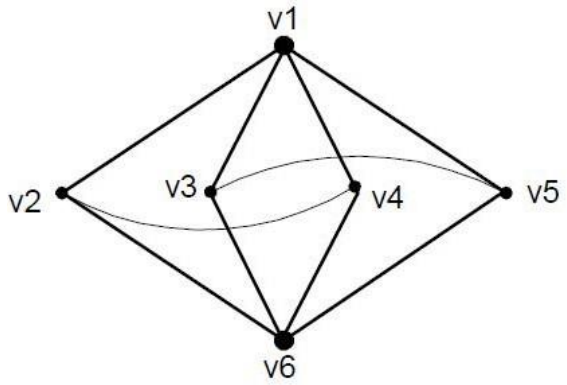
Simpul	B	D	A	E	F	C
Derajat	4	3	2	2	2	1
Warna	x	y	z	y	z	x

(G) = 3, Jadi jumlah minimum ruangan yang dibutuhkan untuk menyimpan semua zat kimia tersebut adalah 3 ruangan.

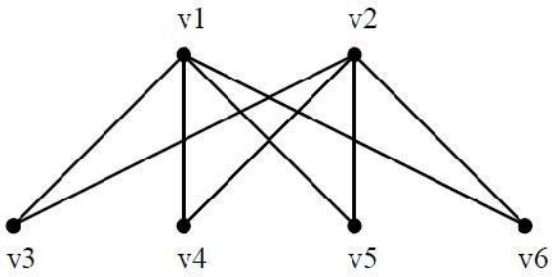
C. SOAL LATIHAN/TUGAS

1. Tentukan pewarnaan graf-graf berikut ini dengan menggunakan algoritma Welch-Powell dan tentukan bilangan kromatiknya.

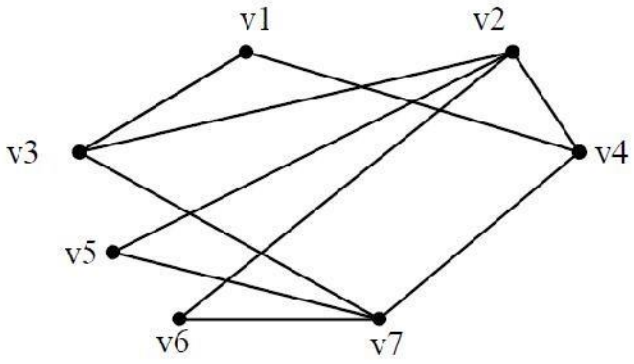
   a. Graf G1



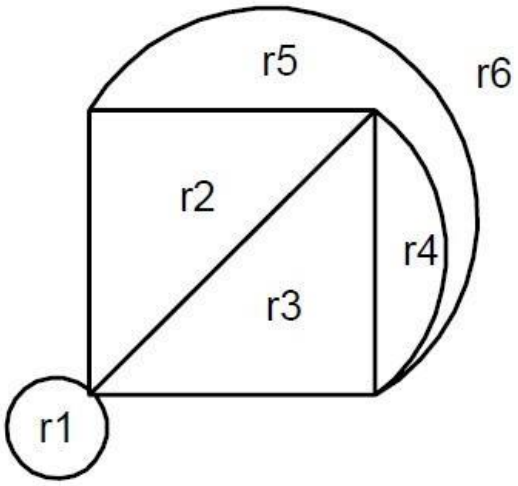
   b. Graf G2



   c. Graf G3



2. Tentukan pewarnaan region pada graf berikut.



D. REFERENSI

PERTEMUAAN 7

PROYEKSI GEOMETRI BIDANG

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan proyeksi geometri bidang.

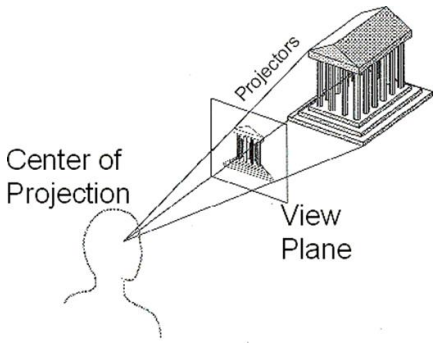
Pada pertemuan ini akan di jelaskan mengenai :

1. Pengertian Proyeksi Geomentri
2. Traksonomi Proyeksi Geomentri
3. Proyeksi Paralel

B. URAIAN MATERI

1. Pengertian Proyeksi Geometri

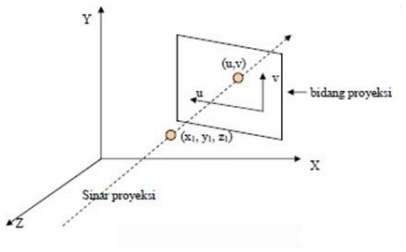
Proyeksi merupakan sebuah jenis transformasi, yaitu transforrmasi kordinat. Proyeksi adalah proses di mana informasi tentang suatu titik dalam sistem kordinat berdimensi-n ditransfer ke sistem koordinat dengan ukuran kurang dari n. Misalnya sebuah titik (x, y, z) pada sistem koordinat 3D dipetakan ke sistem koordinat 2D sehingga menjadi (x, y), maka transformasi harus memperhitungkan pengaruh z pada titik (x, y). Gambar di bawah ini adalah proyeksi.



Gambar 7. 1 Proyeksi Geometri Bidang

Proyeksi dapat dibuat pada bidang datar (planar) atau melengkung. Pada materi ini kita akan membahas tentang proyeksi ke bidang yang disebut bidang Proyeksi

geometri bidang dilakukan melalui sinar proyeksi, yang meninggalkan pusat proyeksi melalui setiap titik benda dan memotong bidang proyeksi untuk mendapatkan bidang proyeksi. Proyeksi secara matematis dapat dijelaskan sebagai berikut:

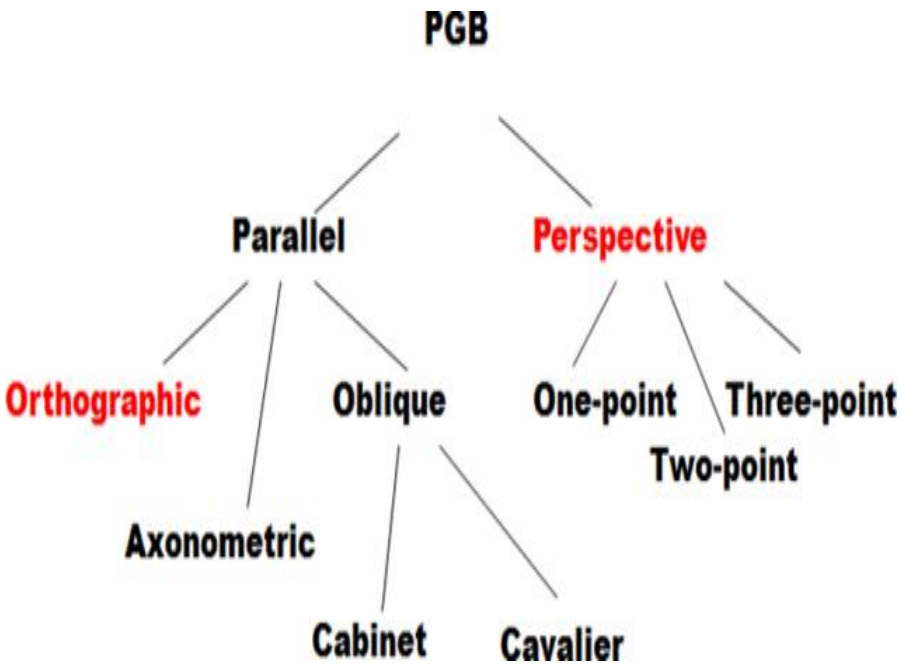


Gambar 7. 2 Proyeksi dan Bidang Proyeksi

2. Taksonomi Proyeksi Geometri

Proyeksi geometri lapangan (PGL) dapat dibagi menjadi dua jenis: proyeksi parelel dan proyeksi perspektif. Perbedaan antara kedua proyeksi ini adalah pada proyeksi perspektif, jarak antara titik pusat proyeksi dan bidang proyeksi adalah berhingga (pasti), sedangkan pada proyeksi paralel jarak antara pusat proyeksi dan bidang proyeksi tidak terhingga.

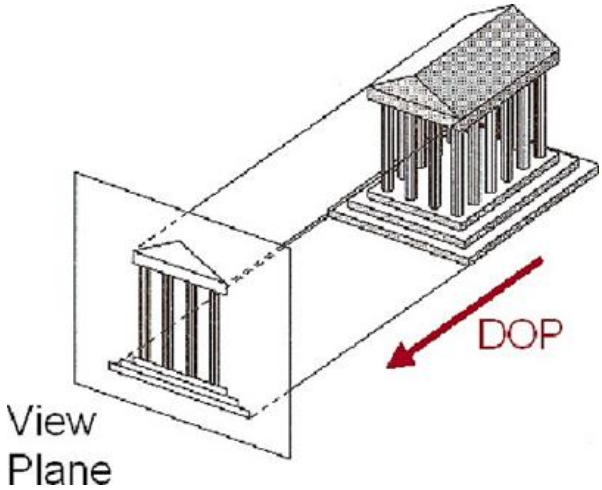
.



Gambar 7. 3 Taksonomi Proyeksi Geometri Bidang

3. Proyeksi Paralel

Proyeksi paralel adalah jenis proyeksi yang berpusat pada tak terhingga. Dengan demikian, arah proyeksi (DOP) sama untuk semua titik. Ilustrasi berikut menunjukkan proyeksi paralel.

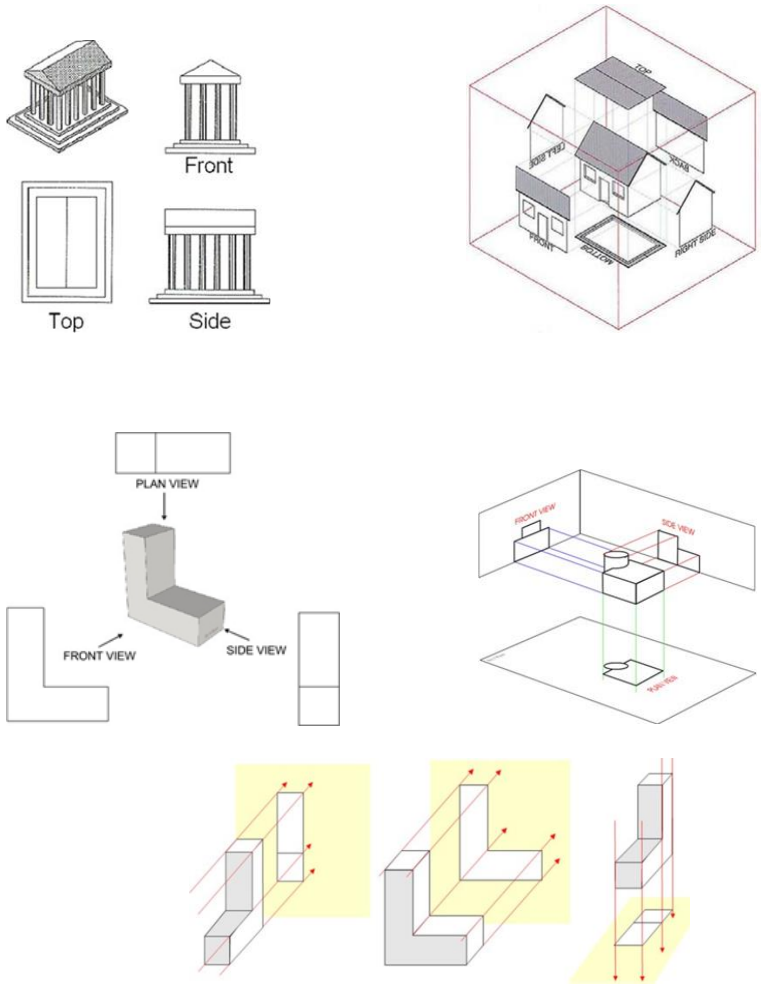


Gambar 7. 4 Proyeksi Paralel

Proyeksi paralel dapat dikategorikan menurut hubungan antara arah proyeksi dan vektor normal bidang proyeksi menjadi dua jenis proyeksi, yaitu ortogonal dan miring.

a. Proyeksi *Orthographic*

Proyeksi ortografik diperoleh jika sinar proyeksi tegak lurus dengan bidang proyeksi. Proyeksi ortografik sering digunakan untuk memperoleh tampilan depan, tampilan atas suatu objek, atau yang disebut dengan citra ortografik multi-tampilan. Tampak atas, tampak belakang, dan tampak samping suatu objek sering disebut sebagai elevasi. Sedangkan tampilan atas dikenal sebagai tampilan atas. Gambar berikut menggambarkan proyeksi ortografik.



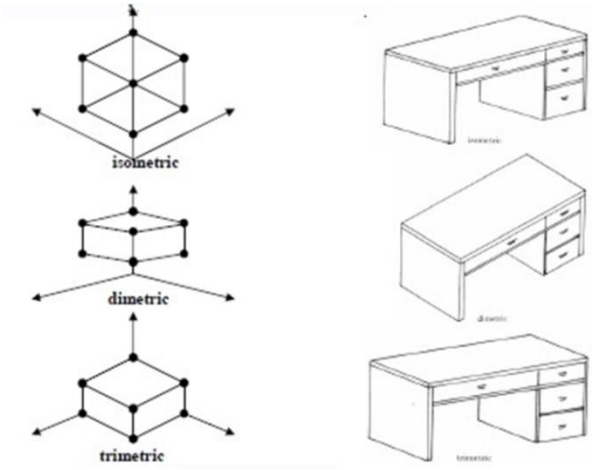
Gambar 7. 5 Bermacam-macam Proyeksi Orthographic

b. Proyeksi *Axonometric*

Proyeksi ortografik yang menunjukkan lebih dari satu permukaan suatu objek disebut proyeksi aksonometri. Jika proyeksi aksonometri dilakukan dengan memposisikan bidang proyeksi sehingga memotong ketiga sumbu koordinat (sumbu mayor) pada sudut yang sama, maka didapatkan proyeksi isometrik.

Jenis proyeksi aksonometri lainnya adalah proyeksi dimetrik, yaitu proyeksi yang diperoleh dengan memposisikan bidang proyeksi sehingga memotong dua sumbu utama pada sudut yang sama, sedangkan proyeksi

trimetrik diperoleh bila tiga sumbu utama memotong bidang iproyeksi pada sudut yang berbeda. sudut. Ilustrasi berikut memperlihatkan tampilan isometrik, diametrik, dan trimetrik.

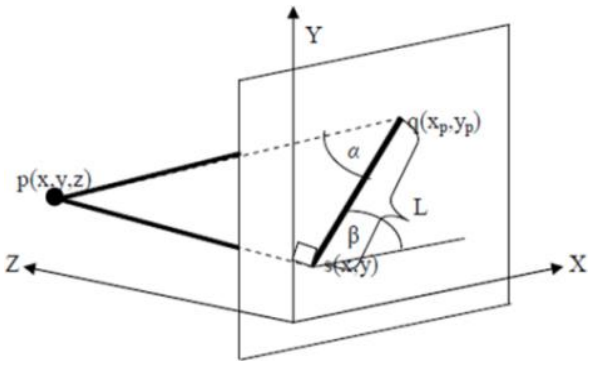


Gambar 7. 6 Proyeksi Isometric, Dimetric dan Trimetric

c. Proyeksi *Oblique*

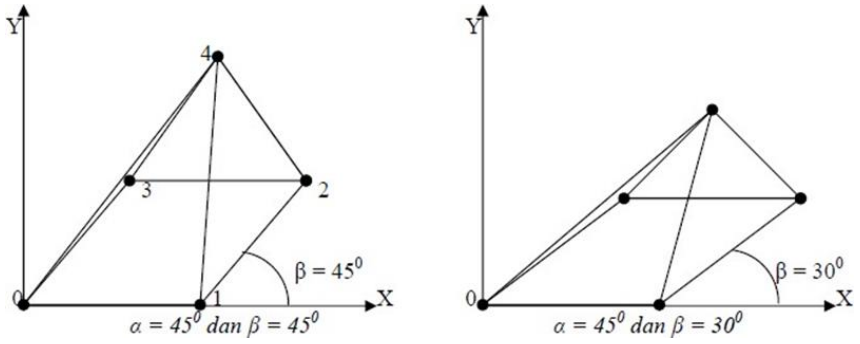
Proyeksi miring dicapai karena balok proyeksi tidak tegak lurus dengan bidang proyeksi. Untuk memproyeksikan bidang miring, dibutuhkan dua sudut yaitu a dan þ, seperti yang ditunjukkan pada gambar 7.7.

Titik (x, y, z) diproyeksikan sebagai titik q (xp, yp) pada bidang proyeksi. Titik yang dihasilkan dari proyeksi iortografi adalah pada s (x, y). Berkas proyeksi terletak pada sudut a terhadap garis q-s yang terletak pada bidang proyeksi. Garis lurus q-s dengan panjang L membentuk sudut terhadap arah horizontal bidang proyeksi.



Gambar 7. 7 Proyeksi Oblique

Gambar berikut ini mengilustrasikan hasil proyeksi *oblique jenis cavalier* dan *cabinet.*

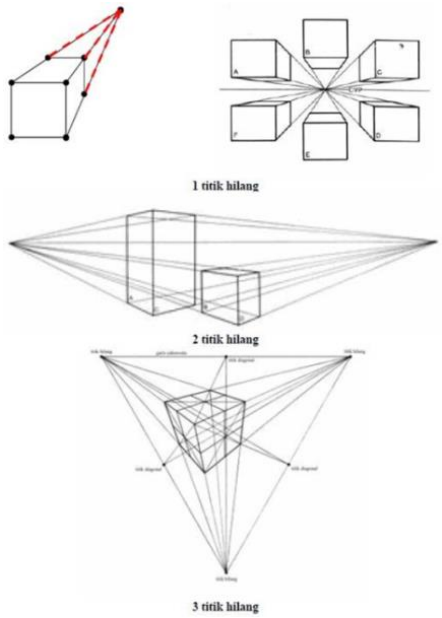


Gambar 7. 8 Proyeksi Oblique Jenis Cavalier dan Cabinet

4. Proyeksi Perspektif

Proyeksi perspektif memberikan perspektif yang lebih realistis daripada proyeksi ortografi. Proyeksi perspektif memberikan pandangan yang sama seperti yang kita lihat setiap hari, karena sebenarnya jarak suatu benda dari kita mempengaruhi kemunculannya. Objek yang tampak jauh akan tampak kecil, dan objek yang dekat akan tampak lebih besar. Efek ini disebut penyusutan.

Dalam proyeksi perspektif, semua garis menghilang pada satu atau lebih titik yang sama, atau ini disebut titik hilang. Hal ini menyepertemuankan kesejajaran tampak tidak sejajar saat memproyeksikan ke dalam perspektif. Bergantung pada tempat kita melihat objek, kita mendapatkan efek: 1 titik hilang, 2 titik hilang, dan 3 titik hilang. Gambar berikut menunjukkan objek sesuai dengan jumlah titik hilang.



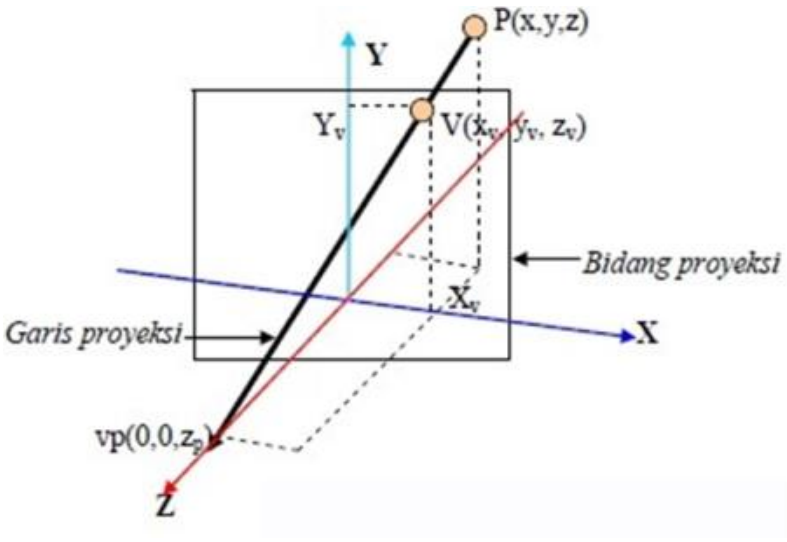
Gambar 7. 9 Proyeksi Perspektif

Perspektif dengan 1 titik hilang akan diperoleh jika tinggi pengamat relatif sama dengan tinggi objek yang diamati dan berada pada jarak yang relatif dekat, dan perspektif dengan 2 titik hilang akan idiperoleh jika pengamat sedikit lebih tinggi atau lebih rendah. dan agak jauh dari subjek saat perspektif tiga titik hilang. akan diterima bila posisi pengamat jauh lebih tinggi atau lebih rendah dari objek yang terlihat.

Table 7. 1 Proyeksi Paralel VS Proyeksi Perspektif

Proyeksi Perspektif	Proyeksi Paralel
• Ukuran berdasarkan jarak – lebih realistik • Jarak dan sudut tidak selalu *preserved* • Garis paralel tidak selalu sejajar	• Baik untuk pengukuran yang membutuhkan ketelitian/presisi • Garis paralel tetap sejajar • Sudut tidak *preserved* • Kurang realistik

Bagaimana proyeksi perspektif terjadi, ditunjukkan pada gambar berikut ini.



Gambar 7. 10 Mekanisme Proyeksi Perspektif

Titik p (x, y, z) diproyeksikan ke bidang x-y melalui garis proyeksi yang memotong sumbu z pada jarak zp. Garis proyeksi akan memotong bidang proyeksi pada titik v (xv, yv, zv).

a. Operasi Proyeksi Perspektif

Proses komputasi untuk operasi proyeksi perspektif ditentukan oleh elemen p, q, r dari matriks transformasi umum tiga dimensi yang diberikan pada pertemuan sebelumnya.

$$\begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ l & m & n & s \end{bmatrix}$$

p, q, r - nilai prediksi, yang nilainya dihitung sebagai

- p = − 1/x untuk pusat sumbu x
- q = − 1/y untuk pusat sumbu y
- r = − 1/z untuk pusat sumbu z

C. SOAL LATIHAN/TUGAS

D. REFERENSI

PERTEMUAN 8

TRANSFORMASI, TRANSLASI DAN ROTASI

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu mengaplikasikan transformasi, translasi, dan rotasi.

Pada pertemuan ini akan dijelaskan mengenai :

1. Pengertian Tranformasi
2. Translasi
3. Rotasi

B. URAIAN MATERI

1. Pengertian Tranformasi

Grafik komputer adalah bidang yang diminati banyak orang. Salah satu subbagian dari grafik komputer adalah pemodelan objek. Pada pemodelan objek dua dimensi (2D), ditemukan berbagai objek yang apat dimodelkan sesuai dengan kondisi tertentu, dan objek yang dimodelkan perlu dimodifikasi. Modifikasi objek ini dapat diselesaikan dengan melakukan berbagai operasi fungsional atau operasi transformasi geometris. Transformasi tersebut dapat berupa transformasi dasar atau kombinasi dari berbagai transformasi geometris.

Contoh transformasi geometris adalah translasi, penskalaan, rotasi, inversi, geser dan peracikan. Transformasi ini disebut transformasi affine. Pada dasarnya, transformasi ini untuk memindahkan objek tanpa merusaknya.

Tujuan dari konversi ini adalah:

a. Merubah atau menyesuaikan komposisi pemandangan
b. Memudahkan membuat objek yang simetris
c. Melihat iobjek dari sudut pandang yang berbeda
d. Memindahkan satu atau beberapa objek dari satu tempat ke tempat lain, ini biasa dipakai untuk animasi komputer.

Proses transformasi dilakukan dengan cara mengalikan matriks objek dengan matriks transformasi untuk menghasilkan matriks baru yang berisi koordinat objek transformasi.

Sebagai contoh, apabila sebuah garis yang melalui titik A(0,1) dan titik B(2,3) ditransformasikan dengan matriks

□ □ □1 2□

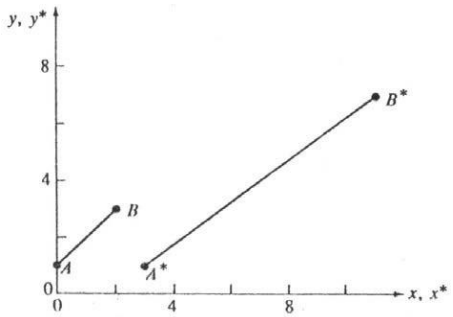
□

maka hasilnya adalah

□L□□T □ □ □0 1□ □1 2□ □ □ 3 1□ □ □L* □

□ □ □

Secara visual proses transformasi ini dapat dilihat pada gambar berikut:



Gambar 8. 1 Ilustrasi Transformasi Sebuah Garis

2. Translasi

Transformasi translasi adalah operasi yang menyepertemuankan objek 2D berpindah dari satu posisi ke posisi lain. Perubahan ini terjadi pada arah yang sejajar dengan sumbu X dan Y, dan translasi diselesaikan dengan menambahkan translasi pada titik koordinat dengan vektor translasi (tx, ty), dimana tx adalah translasi menurut sumbu x dan ty adalah di sepanjang sumbu y Terjemahan. Koordinat baru dari titik terjemahan dapat ditemukan dengan menggunakan rumus berikut:

x' = x + tx (x,y) = titik asal sebelum translasi

y' = y + ty (x',y') = titik baru hasil translasi

Terjemahan adalah transformasi dengan bentuk tetap, yaitu menggerakkan objek sebagaimana adanya. Setiap titik benda akan diterjemahkan dengan ukuran yang sama.

Dalam operasi penerjemahan, setiap titik dalam entitas terjemahan bergerak dengan jarak yang sama. Gerakan ini hanya dapat berpengaruh pada arah X atau hanya pada arah Y atau di kedua arah pada saat yang bersamaan.

Terjemahan juga berlaku untuk garis, objek, atau kombinasi objek 2D lainnya. Untuk tujuan ini, setiap titik pada garis lurus atau objek yang diterjemahkan ke arah x dan y masing-masing adalah tx, ty.

Contoh:

Untuk mengilustraskan suatu objek berupa segitiga dengan koordinat A(10,10) B(30,10) dan C(10,30) dengan tx,ty(10,20), tentukan kordinat yang barunya !

Jawab

A : x' = 10 + 10 = 20

y' = 10 + 20 = 30

A' = (20,30)

B : x' = 30 + 10 = 40

y' = 10 + 20 = 30

B' = 40,30)

C : x' = 10 + 10 = 20

y' = 30 + 20 = 50

B' = (20,50)

3. Rotasi

Rotasi adalah operasi memindahkan suatu objek pada titik tengah atau sumbu rotasi yang dipilih berdasarkan sudut rotasi tertentu. Untuk melakukan rotasi, diperlukan sudut putar dan titik pivot (xp, yp) dari benda yang akan diputar.

Rotasi biasanya dilakukan pada titik tertentu ke sumbu tertentu, seperti sumbu x, sumbu y, atau garis tertentu yang sejajar dengan salah satu sumbu ini. Titik referensi rotasi bisa berada di titik tengah atau di titik lain. Biasanya dalam geometri, jika Anda memutar searah jarum jam, nilai sudutnya negatif. Sebaliknya jika berputar berlawanan arah jarum jam maka nilai sudutnya positif.

Rotasi dapat dinyatakan sebagai:

$$x' = r\cos(\phi + \theta) = r\cos\phi \quad \text{os } \theta \quad - r\sin\phi \quad \text{in } \theta$$

$$y' = r\sin(\phi + \theta) = r\cos\phi \quad \sin\theta \quad + r\sin\phi \quad \text{os } \theta$$

sedangkan diketahui

$$x = r\cos\phi \ , \ y = r\sin\phi$$

lakukan subtitusi, maka :

$$x' = x\cos\phi - y\sin\theta$$

$$y' = x\sin\phi + y\cos\theta$$

Matriks rotasi dinyatakan dengan :

$$P' = R.P$$

$$R = \begin{pmatrix} \cos 8 & -\sin 8 \\ \sin 8 & \cos 8 \end{pmatrix}$$

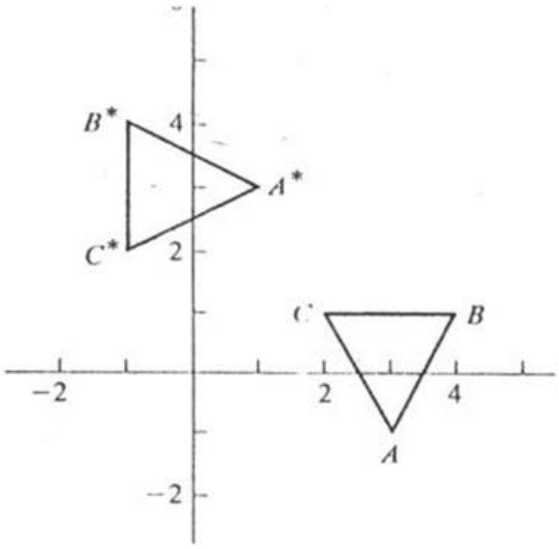
Rotasi suatu titik terhadap pivot point (xp,yp) :

$$x' = xp+(x - xp) \cos \theta - (y - yp) \sin \theta$$

$$y' = yp+(x - xp) \sin \theta + (y - yp) \cos \theta$$

Contoh 1:

Anda akan mendapatkan koordinat titik-titik yang membentuk segitiga ((3, -1), (4, 1), (2, 1). Gambarkan objek, lalu gambar objek baru, seperti memutar objek lama Objek 90 derajat berlawanan arah jarum jam Rotasi, pusat rotasinya adalah (0,0). Oleh karena itu, dengan mengalikan titik-titik segitiga dengan matriks transformasi yang sesuai, hasilnya digambarkan sebagai berikut:



Gambar 8. 2 Ilustrasi Rotasi

Contoh 2:

Gunakan bentuk segitiga dari koordinat A (10,10), B (30,10) dan C (10,30) untuk mendeskripsikan rotasi objek relatif terhadap titik pusat Cartesian (sudut rotasi 10 adalah 300 °), 10), yaitu dengan menghitung koordinat rotasi setiap titik satu per satu.

Jawab:

Titik A

x’= xp+(x - xp) cos 8 - (y - yp) sin 8

=10+(10-10)*0.9 – (10-10)*0.5 = 10

y’= yp+(x - xp) sin 8 + (y - yp) cos 8

= 10+(10-10)*0.5 – (10-10)*0.9 = 10

Titik A’(10,10)

Titik B

x’= xp+(x - xp) cos 8 - (y - yp) sin 8

=10+(30-10)*0.9 – (10-10)*0.5 = 28

y’= yp+(x - xp) sin 8 + (y - yp) cos 8

= 10+(30-10)*0.5 – (10-10)*0.9 = 20

C. SOAL LATIHAN\TUGAS

D. REFERENSI

PERTEMUAN 9

DILATASI

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu mengaplikasikan dilatasi.

Pada pertemuan ini akan dijelaskan mengenai :

1. Pengertian Dilatasi
2. Shear

B. URAIAN MATERI

1. Pengertian Dilatasi

Dilatasi adalah transformasi yang membuat mirror (pencerminan) dari image suatu objek. Image mirror untuk dilitasi 2D dibuat relatif terhadap sumbu dari dilitasi dengan memutar 180o terhadap dilitasi. Sumbu dilitasi dapat dipilih pada bidang x,y. Dilitasi terhadap garis y=0, yaitu sumbu x dinyatakan dengan matriks.

$$R = \begin{pmatrix} \cos 8 & -\sin 8 \\ \sin 8 & \cos 8 \end{pmatrix}$$

Transformasi membuat nilai x sama tetapi membalikan nilai y berlawanan dengan posisi koordinat Langkah :

a. Objek diangkat
b. Putar 180o terhadap sumbu x dalam 3D
c. Letakkan pada bidang x,y dengan posisi berlawanan
d. Refleksi terhadap sumbu y membalikan koordinat dengan nilai y tetap.

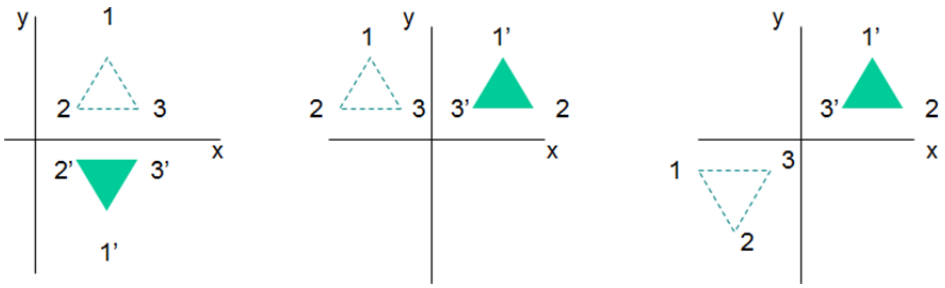
$$\begin{pmatrix} -1 & 0 & 0 \end{pmatrix}$$

0 1 0

Dilitasi terhadap sumbu x dan y sekaligus dilakukan dengan dilitasi pada sumbu x terlebih dahulu, hasilnya kemudia direfleksi terhadap sumbu y. Transformasi ini dinyatakan dengan :

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Dilitas ini sama dengan rotasi 180° pada bidang xy dengan koordinat menggunakan titik pusat koordinat sebagai pivot point suatu objek terhadap garis y=x dinyatakan dengan bentuk matriks.Ilustrasi proses dilitasi pada sumbu-sumbu utama digambarkan pada gambar berikut:



Gambar 9. 1 Ilustrasi Dlitasi

Matriks dapat diturunkan dengan menggabungkan suatu sekuen rotasi dari sumbu koordinat merdilitasi matriks. Pertama-tama dilakukan rotasi searah jarum jam dengan sudut 45° yang memutar garis y=x terhadap sumbu x. Kemudian objek direfleksi terhadap sumbu y, setelah itu objek dan garis y=x dirotasi kembali ke arah posisi semula berlawanan arah dengan jarum jam dengan sudut rotasi 90°

Untuk mendapatkan refleksi terhadap garis y=-x dapat dilakukan dengan tahap :

a. Rotasi 45° searah jarum jam

b. Dilitasi terhadap axis y

c. Rotasi 90° berlawanan arah dengan jarum jam

d. Dinyatakan dengan bentuk matriks

Dilitasi terhadap garis y=mx+b pada bidang xy merupakan kombinasi transformasi translasi – rotasi – dilitasi .

a. Lakukan translasi mencapai titik perpotongan koordinat

b. Rotasi ke salah satu sumbu

c. Dilitasi objek menurut sumbu tersebut

2. Shear

Shear adalah bentuk transformasi yang membuat distorsi dari bentuk suatu objek, seperti menggeser sisi tertentu. Terdapat dua macam shear yaitu shear terhadap sumbu x dan shear terhadap sumbu y.

Shear terhadap sumbu x

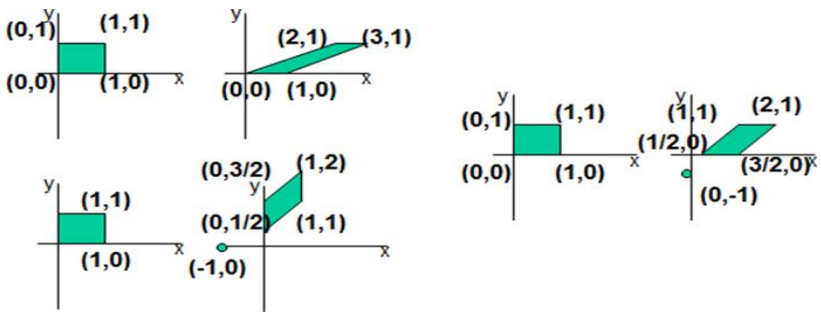
$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Dengan koordinat transformasi

x'=x y'= shy.x+y

Parameter shy dinyatakan dengan sembarang bilangan. Posisi koordinat kemudian menurut arah vertikal

Gambar di bawah mengilustrasikan proses shearing.



Gambar 9. 2 Ilustrasi Proses Shearing

C. SOAL LATIHAN/TUGAS

D. REFERENSI

PERTEMUAN 10

VIEWING DAN CLIPPING

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu mengaplikasikan viewing dan clipping.

Pada pertemuan ini akan dijelaskan mengenai :

1. Viewing
2. Clipping

B. URAIAN MATERI

1. Viewing

Sekarang kita mempertimbangkan mekanisme formal untuk merender representasi gambar ke perangkat keluaran. Biasanya, paket grafis memungkinkan pengguna untuk menentukan bagian mana dari pi & ke yang harus ditampilkan dan di mana bagian itu harus ditempatkan pada perangkat tampilan. Setiap sistem koordinat Cartesian yang nyaman yang disebut kotak referensi koordinat dunia dapat digunakan untuk menentukan gambar. Untuk gambar dua dimensi, tampilan dipilih dengan menentukan sebagian area dari total area gambar. Seorang pengguna dapat memilih satu area untuk ditampilkan, atau beberapa area dapat dipilih untuk tampilan simultan atau untuk urutan pan animasi pada sebuah adegan. Bagian gambar dalam area yang dipilih dipetakan ke area tertentu dari koordinat perangkat. Jika beberapa area tampilan dipilih, area ini dapat ditempatkan di lokasi tampilan terpisah, atau beberapa area dapat ditempatkan di area tampilan lain yang lebih besar. Transformasi dari koordinat dunia ke koordinat perangkat mencakup operasi terjemahan, putar, dan skala, serta prosedur untuk menghapus bagian gambar yang berada di luar batas area tampilan yang dipilih.

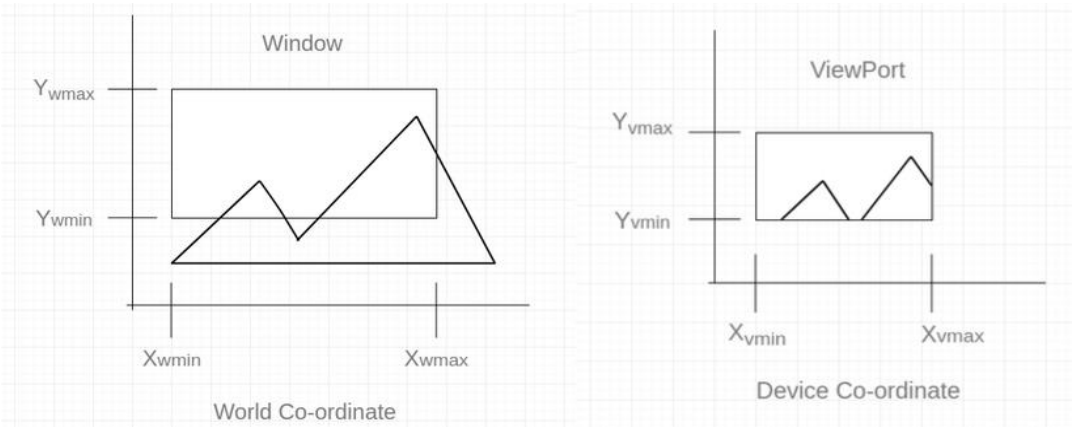
a. Struktur Viewing :

1) Window : Area koordinat dunia (world-coordinate) yang dipilih untuk ditampilkan. Untuk mendefinisikan apa yang ditampilkan.
2) Viewport : Area dari display device (misal:monitor) dimana window dipetakan. Untuk mendefinisikan di mana harus ditampilkan.
3) Viewing Transformation : Pemetaan bagian dari koordinat dunia ke koordinat device.

b. Window

Transformasi Window to Viewport adalah proses mengubah objek koordinat dunia 2D menjadi koordinat perangkat. Objek di dalam world atau clipping window clipping dipetakan ke area pandang yang merupakan area di layar tempat koordinat dunia dipetakan untuk ditampilkan.

World Coordinate Area yang dipilih untuk tampilan disebut Window. Sebagian besar window berbentuk persegi panjang.



Gambar 10. 1 Window

c. Viewing Transformation

Transformasi terakhir yang diterapkan pada objek sebelum ditampilkan dalam gambar adalah transformasi window-to-viewport, yang memetakan jendela tampilan persegi pada bidang xy yang berisi pemandangan ke kisi piksel persegi panjang tempat gambar akan ditampilkan. Saya akan berasumsi di sini bahwa jendela tampilan tidak diputar; bahwa itu, sisi-sisinya sejajar dengan sumbu x dan y. Dalam hal ini, transformasi window-to-viewport dapat diekspresikan dalam bentuk terjemahan dan transformasi penskalaan.

Mari kita lihat kasus umum di mana area pandang memiliki koordinat piksel mulai dari 0 di kiri hingga lebar di kanan dan dari 0 di atas hingga tinggi di bawah. Dan asumsikan bahwa batas pada jendela tampilan adalah kiri, kanan, bawah, dan atas. Dalam hal ini, transformasi window-to-viewport dapat diprogram sebagai:

```
scale( width / (right-left), height / (bottom-top) );
translate( -left, -top )
```

Ini harus menjadi transformasi terakhir yang diterapkan ke suatu titik. Karena transformasi diterapkan ke titik-titik dengan urutan kebalikan dari yang ditentukan dalam program, transformasi tersebut harus menjadi transformasi pertama yang ditentukan dalam program. Untuk melihat bagaimana ini bekerja, pertimbangkan satu titik (x, y) di jendela tampilan. (Titik ini berasal dari beberapa objek dalam pemandangan. Beberapa transformasi pemodelan mungkin telah diterapkan pada objek untuk menghasilkan titik (x, y), dan titik tersebut sekarang siap untuk transformasi terakhirnya menjadi koordinat area pandang.) Koordinat (x, y) pertama kali diterjemahkan oleh (-left, -top) menjadi (x-left, y-top). Koordinat ini kemudian dikalikan dengan faktor skala yang ditunjukkan di atas, sehingga menghasilkan koordinat akhir

```
x1 = width / (right-left) * (x-left)
y1 = height / (bottom-top) * (y-top)
```

Perhatikan bahwa titik (kiri, atas) dipetakan ke (0,0), sedangkan titik (kanan, bawah) dipetakan ke (lebar, tinggi), yang kita inginkan. Masih ada pertanyaan aspek rasio. Seperti disebutkan di Subbagian 2.1.3, jika kita ingin memaksa rasio aspek jendela agar sesuai dengan rasio aspek viewport, mungkin perlu menyesuaikan batas pada jendela. Berikut adalah pseudocode untuk subrutin yang akan melakukannya, dengan asumsi lagi bahwa pojok kiri atas viewport memiliki koordinat piksel (0,0):

```
subroutine applyWindowToViewportTransformation (
          left, right,   // horizontal limits on view window
          bottom, top,   // vertical limits on view window
          width, height, // width and height of viewport
          preserveAspect // should window be forced to match viewport aspect?
       )

   if preserveAspect :
       // Adjust the limits to match the aspect ratio of the drawing area.
       displayAspect = abs(height / width);
       windowAspect = abs(( top-bottom ) / ( right-left ));
       if displayAspect > windowAspect :
           // Expand the viewport vertically.
           excess = (top-bottom) * (displayAspect/windowAspect - 1)
           top = top + excess/2
           bottom = bottom - excess/2
       else if displayAspect < windowAspect :
           // Expand the viewport vertically.
           excess = (right-left) * (windowAspect/displayAspect - 1)
           right = right + excess/2
```

d. Viewing Transformation

Titik *P (x, y)* dan *P0(x_0, y_0)* yang bertindak sebagai vector arah pandang atas bagi pengamat (viewer) atau sebagai sumbu Y_v dari viewing coordinates system, vector satuan v pada arah Y_v pada viewing coordinat system. Rumusnya sebagai berikut :

$$v = \frac{p-p_0}{|p-p_0|} = \frac{(x-x_0\,,\,y-y_0)}{\sqrt{(x-x_0)^2+(y-y_0)^2}} = (v_x, v_y)$$

Vector satuan *u* pada coordinates system dihitung dengan cara perkalian silang antara vector satuan v dengan sumbu z pada viewing coordinates system:

$$\boldsymbol{u = v * (0, 0, 1)} \textbf{ atau } \boldsymbol{u = (v_y, -v_x)}$$

Matriks transformasi M dari World Coordinates System ke Viewing Coordinates System dinyatakan sebagai komposisi matriks berikut :

$$M = R * T(-x_0, -y_0)$$

$$M = \begin{pmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Sehingga diperoleh titik terhadap viewing Coordinates System adalah *P* viewer = M * *P* word

Keterangan : *P*word = Posisi titik terhadap World Coordinat System

*P*viewer = Posisi titik terhadap Viewig Coordinat System

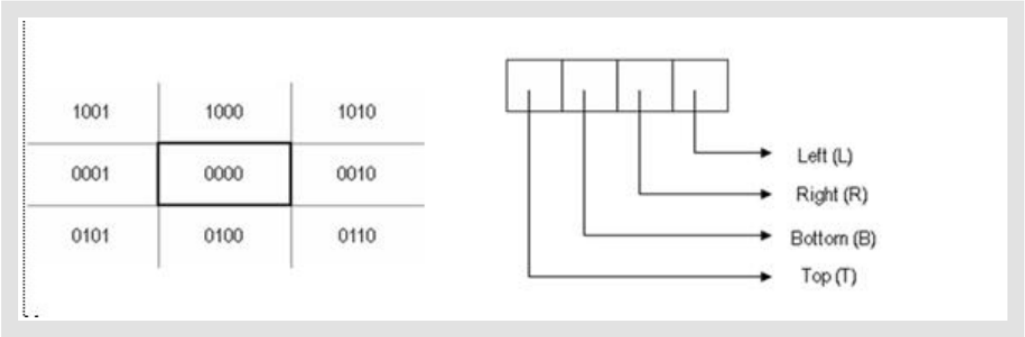
2. Clipping

Metode clipping adalah metode yang digunakan untuk menentukan garis yang perlu digambar atau tidak. Alasan dilakukanna clpping adalah untuk menghindari perhitungan koordinat pixel yang rumit dan interpolasi parameter. Clpping dilakuakn sebelum proses rasterization. Setelah proses clipping selanjutnya dilakukan proses rasterization yang mana dilakukan pengkonversian suatu citra vektor ke citra bitmap.

Tiap metode mempunya beberapa algoritma dan tentunya tiap algoritma memiliki kelebihan dan kekurangan untuk dianalisis. Contohnya pada algoritma clipping didapatkan algoritma Liang-Barsky yang terbaik karena kecepatan waktu yang efisien dan juga stabil. Untuk metode rasterization didapat algoritma Midpoint yang terbaik karena operasi bilangan pada Midpoint dilakukan dengan cara menghilangkan operasi bilangan riel dengan bilangan integer yang mana bilangan integer jauh lebih cepat dibandingkan dengan operasi bilangan riel. Oleh karena itu, komputasi midpoint lebih cepat delapan kali pada pembuatan garis lurus dan lima belas kali pada penggambaran lingkaran. Sedangkan pada metode Hidden Surface Remove, algoritma yang terbaik adalah algoritma scan Line karena pada algoritma ini menggunakan memori yang lebih sedikit dan dari segi kecepatan juga lebih unggul.

a. Point Clipping

Clipping line bisa ditentukan dengan beberapa alghoritma, tapi untuk kasus ini saya menggunakan algoritma Cohen-Suterland. Berikut ini adalah gambar pembagian dari baris segmen window dengan algoritma Cohen-Sutherland.



Keterangan :

L :	1 apabila x < xmin0 apabila x ≥ xmin	B :	1 apabila y < ymin0 apabila y ≥ ymin
R :	1 apabila x > xmax0 apabila x ≤ xmax	T :	1 apabila y > ymax0 apabila y ≤ ymax

Gambar 10. 2 Pembagian dari baris segmen window Sutherland

Dengan asumsi bahwa jendela klip adalah persegi panjang dalam posisi standar, kami menyimpan titik P = (x, y) untuk tampilan jika pertidaksamaan berikut terpenuhi:

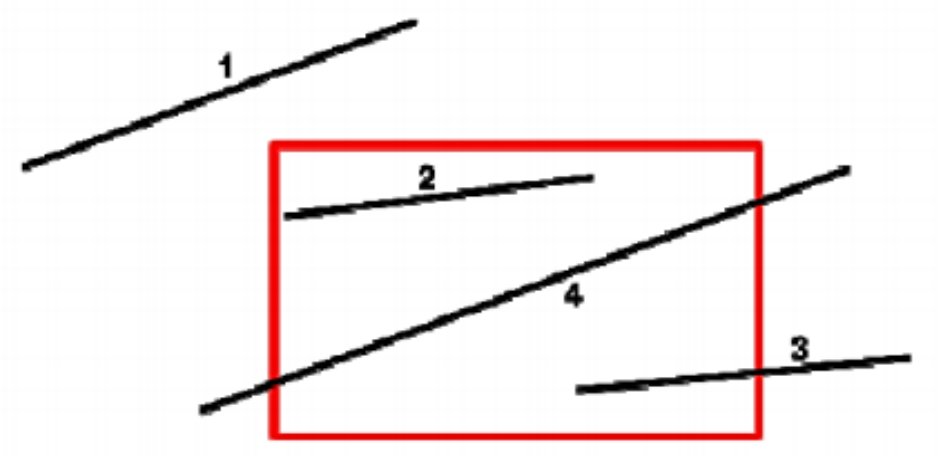
$$xw_{\min} \leq x \leq xw_{\max}$$

$$yw_{\min} \leq y \leq yw_{\max}$$

Dengan tepi Clipp Window (*xwmin xwmax, ywmin, ywmax*) dapat berupa batas jendela koordinat dunia atau batas area pandang. Jika salah satu dari empat ketidaksamaan ini tidak terpenuhi, poinnya akan dipotong (tidak disimpan untuk ditampilkan).

b. Line Clipping

Line clipping diproses dengan inside-outside test dengan memeriksa endpoint dari garis tersebut.



Gambar 10. 3 Line Clipping

Dari test tersebut dapat dikategorikan sebagai berikut:

1) Garis yang kedua endpoint-nya ada di dalam batas clipping akan disimpan (garis 2)

2) Garis yang kedua endpoint tidak berada dalam batas clipping maka garis tersebut berada di luar window (garis 1).

3) Semua garis lain yang memotong satu atau lebih batas clipping memerlukan algoritma clipping yang dapat mengidentifikasi dengan efisien bahwa garis berada diluar batas clipping window (garis 3 dan 4).

Kondisi garis terhadap clipping window dapat dikategorikan seperti tabel berikut:

Nama	Kondisi
Invisible	Tidak kelihatan, terletak di luar *clipping window*
Visible	Terletak di dalam *clipping window*
Half Partial	Terpotong sebagian oleh *clipping window*
Full Partial	Terpotong penuh oleh *clipping window*, garis melintasi *clipping window*

Untuk kondisi garis yang invisible dan visible tidak perlu dilakukan aksi clipping, dimana untuk:

1) Invisible, tidak perlu ditampilkan

2) Visible, langsung ditampilkan

Persamaan untuk segmen garis dengan endpoint (x1,y1) dan (x2,y2) yang keduanya terletak diluar clipping window adalah:

$$x = x1 + u(x2 - x1)$$
$$y = x1 + u(x2 - x1)$$
$$0<=u<=1$$

Persamaan tersebut digunakan untuk mengenali nilai parameter untuk koordinat pemotongan dengan batas *clipping window*.

1) Membaca data garis

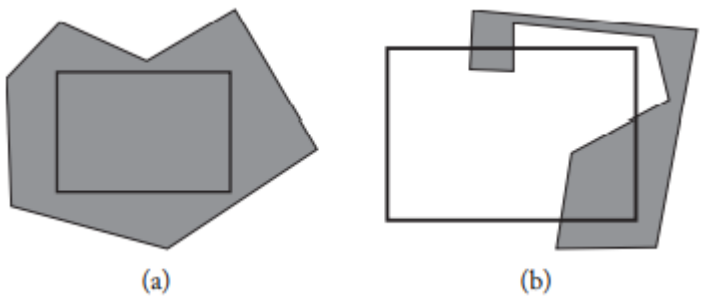
2) Membaca data *clipping window*

3) Mengecek koondisi garis terhadap *clipping window*

4) Proses *clipping*

c. Polygon Clipping

Pemotongan tidak hanya untuk simpul dan tepi poligon. Pada Gambar, semua simpul dan tepi berada di luar persegi panjang pemotongan; akan tetapi, poligon tidak berada di wilayah luar dari persegi panjang: kliping adalah persegi panjang itu sendiri.



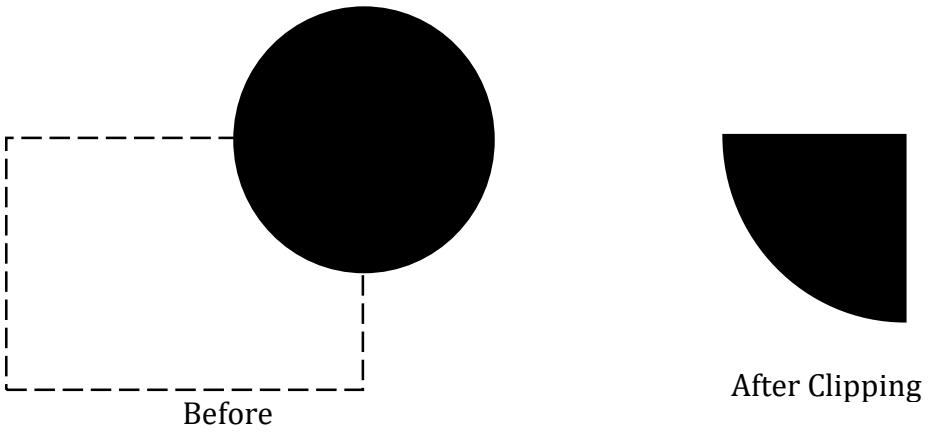
Gambar 10. 4 Polygon Clipping

Dalam hal objek yang akan dipotong adalah sebuah titik, satu-satunya operasi yang harus dilakukan adalah klasifikasi. Jika objek berupa garis lurus, kita juga dapat menghitung persimpangan sebagai cara untuk mengklasifikasikan setiap segmen yang dihasilkan. Dalam kasus ini, kita tidak harus berurusan dengan penataan. Tetapi dalam kasus daerah Polygon, pemotongan melibatkan: persimpangan, klasifikasi, dan penataan.

Secara umum, untuk daerah klip kami melakukan pemotongan batas dan kemudian menentukan kurva Polygon batas dari setiap daerah yang dihasilkan dari operasi pemotongan. Karena pemotongan batas direduksi untuk merekam urutan segmen garis lurus, hal ini tidak menimbulkan kesulitan apa pun — tantangan sebenarnya adalah dalam Polygon data untuk mendapatkan batas setiap wilayah yang dihasilkan dari pemotongan tersebut. Kasus paling sederhana dari pemotongan Polygon adalah pemotongan segitiga, yang akan kita pelajari selanjutnya.

d. Curve Clipping

Area dengan batas melengkung dapat dipotong dengan metode yang mirip dengan yang dibahas di bagian sebelumnya. Prosedur kurva melibatkan persamaan nonlinier, dan ini membutuhkan lebih banyak pemrosesan dari pada untuk objek dengan batas linier.



Gambar 10. 5 Curve Clipping

Persegi panjang pembatas untuk lingkaran atau objek melengkung lainnya dapat digunakan pertama kali untuk menguji tumpang tindih dengan

jendela klip persegi panjang. Jika persegi panjang pembatas untuk objek sepenuhnya berada di dalam jendela, kami menyimpan objek tersebut Jika persegi panjang ditentukan sepenuhnya berada di luar jendela, kami membuang objeknya Dalam kedua kasus tersebut; tidak diperlukan perhitungan lebih lanjut Tetapi jika uji persegi panjang pembatas gagal: kita dapat mencari pendekatan lain yang menghemat komputasi. Untuk lingkaran, kita dapat menggunakan luasan koordinat dari masing-masing kuadran dan kemudian oktan untuk pengujian pendahuluan sebelum menghitung persimpangan kurva-jendela_ Untuk elips, kita dapat menguji luasan koordinat kuadran individu. Gambar 10.5 mengilustrasikan kliping lingkaran pada jendela persegi Panjang Prosedur serupa dapat diterapkan saat memotong objek melengkung ke wilayah klip poligon umum. Pada lintasan pertama, kita dapat memotong persegi panjang pembatas objek terhadap persegi panjang pembatas dari wilayah klip. Jika kedua daerah tumpang tindih, kita perlu menyelesaikan persamaan kurva garis secara simultan untuk mendapatkan titik potong.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 10	1. Jelaskan definisi dari viewing & clipping?

D. REFERENSI

PERTEMUAN 11

ANTIALIASING

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu mengaplikasikan antiliasing.

Pada pertemuan ini akan dijelaskan mengenai Teknik Antialiasing

B. URAIAN MATERI

1. Teknik Antialiasing

Anti-aliasing dalam **pengolahan sinyal digital** adalah teknik mengurangi artifak distorsi dalam merepresentasikan citra resolusi tinggi pada resolusi yang lebih rendah. Artifak distorsi disebut aliasing. Anti-aliasing digunakan dalam fotografi digital, grafik komputer, audio digital, dan bidang lainnya.

Anti-aliasing berarti menghilangkan komponen sinyal yang memiliki frekuensi lebih tinggi dari yang dapat diterima oleh alat perekam (*sampling*). Jika perekaman dilakukan tanpa menghilangkan bagian sinyal ini, maka dapat menyepertemuankan tampilan citra yang tidak diinginkan (*noise*).

Anti-aliasing adalah kata yang sering dilemparkan oleh fotografer dan gamer ketika berhadapan dengan grafik dan gambar. Lihatlah apa yang anti-aliasing, mengapa kita menggunakannya, dan yang paling penting, ketika yang terbaik adalah tidak menggunakannya.

Vektor dan Piksel, dan Mengapa Kamera Mengambil Gambar Dengan Piksel

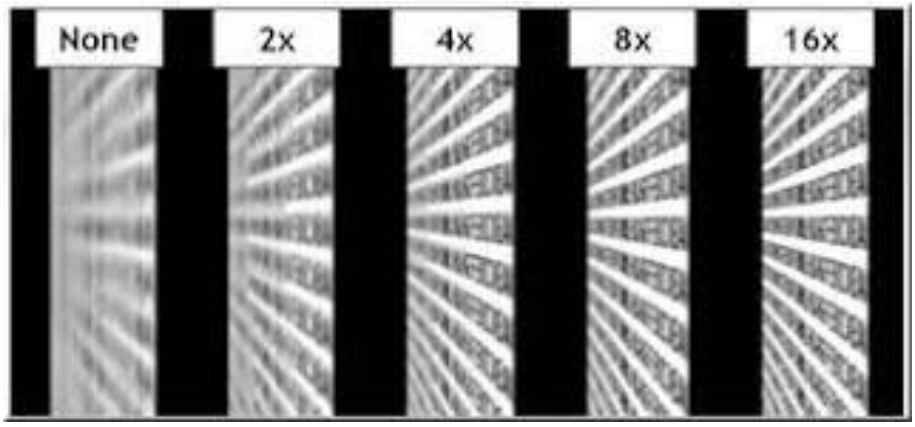
Gambar 11. 1 Vektor dan Piksel

Ada sejumlah perbedaan mendasar antara keduanya: **piksel** adalah susunan cahaya, pigmen, atau warna; **vektor** adalah representasi matematis dari garis, bentuk, gradien, dll. Vektor tepat; mereka ada di koordinat absolut pada grid aljabar. Karena mereka sangat absolut, tidak ada garis kabur antara tempat mereka berada dan di mana mereka tidak. Bahkan jika monitor tidak dapat membuat ketipisan tak terbatas segmen garis (selalu harus menampilkannya dalam piksel), itu masih setipis garis yang hanya ada di dunia matematika teoretis.

Itu masalah dengan fotografi-cahaya tidak setepat mungkin perlu ditangkap dengan cara matematika yang sempurna. Sangat mungkin bahwa bahkan jika kita mengembangkan kamera yang mampu membaca lokasi foton individu dengan presisi kuantum ketika mereka memukul sensor, karena sifat aneh fisika pada tingkat kuantum, partikel individu sebenarnya dapat muncul di beberapa tempat pada sensor di waktu yang sama. Ini berarti sangat tidak mungkin untuk mendapatkan lokasi absolut dari partikel cahaya tunggal pada saat itu mengenai sensor-fotografi hanyalah perkiraan tentang bagaimana cahaya itu ditangkap. Aksi penghentian (kemampuan kamera untuk membuat gambar yang tajam dari objek yang bergerak) tidak pernah bisa menjadi sempurna, setidaknya tampaknya sangat, sangat tidak mungkin.

Line Thinning atau yang biasa disebut Anti Aliasing adalah teknik mengurangi jagged (tepi objek yang bergerigi) pada objek, agar dapat menampilkan resolusi tinggi pada resolusi yang lebih rendah, bisa juga Anti aliasing ini digunakan untuk mengurangi efek aliasing seperti gambar yang

diperbesar akan membentuk sebuah tangga/garis patah-patah. Anti aliasing sejauh ini dibagi menjadi 2x,4x,8x,16x semakin tinggi tingkat aliasing nya maka semakin halus atau semakin menarik gambar/objek tersebut.

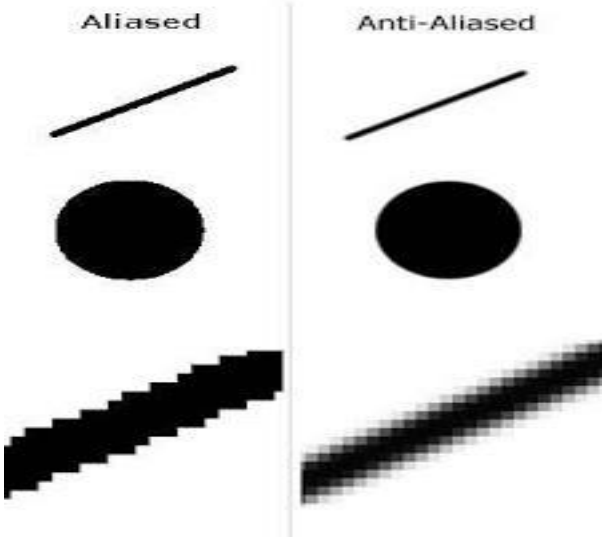


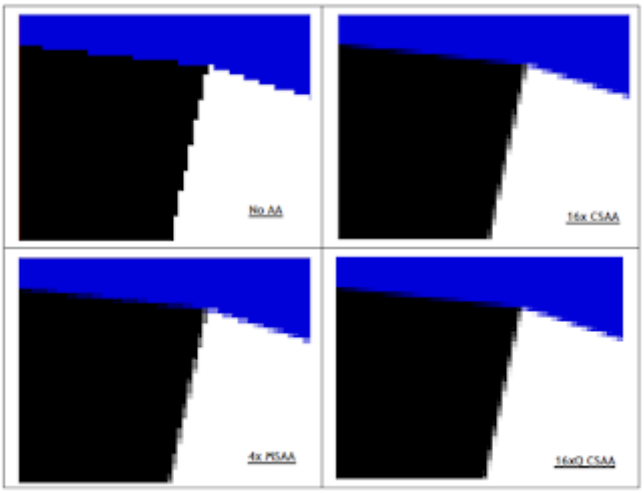
Gambar 11. 2 Line Thinning

Sebagai contoh yang lebih sederhana :

Jika kamu zoom suatu gambar dengan jarak yang sangat dekat, maka kamu tidak akan melihat gambar tersebut seperti semula, melainkan gambar tersebut akan pecah atau bergerigi pada pinggir, itu disepertemuankan karena kamu belum memberikan anti aliasing pada gambar tersebut. Contoh selanjutnya jika kalian membuat garis, dan garis yang anda buat menggunakan ukuran yang kecil maka setelah di zoom akan terlihat seperti patah-patah jika tidak diberikan anti aliasing.

Fungsi dari anti aliasing ini yaitu sebagai filter yang mengubah warna pada pixel disekitar objek yang terlihat patah-patah untuk dibuat menjadi objek yang lebih halus.





Gambar 11. 3 Antialiasing

Jenis- jenis Anti Aliasing

a. FXAA (*Fast Approxiamate Anti Aliasing*) digunakan untuk memproses bingkai (pada pinggir gambar yaitu seperti gambar patah-patah) yang ditampilkan untuk mengurangi aliasing dengan relative cepat, tetapi kelemahannya dapat membuat gambar menjadi kabur/buram.

b. MSAA (*Multisampling Antialiasing*) diterapkan hanya pada bagian pinggir gambar yang terlihat patah-patah, jadi bisa dibilang MSAA ini masih terdapat efek patah-patah nya karena yang diterapkan hanya pada pinggirnya saja. Bukannya pada seluruh frame.

c. TXAA (*Temporal Antialiasing*) ini sama seperti dengan MSAA dengan banyak macamnya. Txaa ini digunakan untuk menyempurnakan

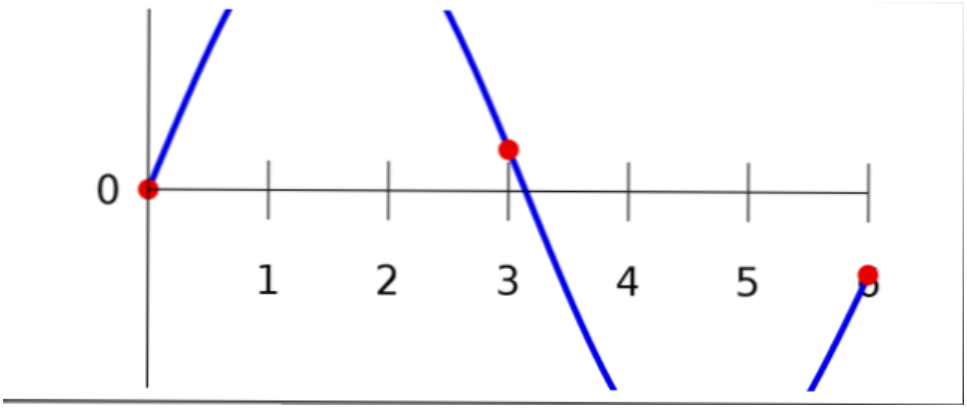
objek/gambar pada pinggir yang akan kita tampilkan sehingga gambar/objek nya terlihat sempurna, tetapi jika kalian menggunakan jenis ini akan membuat gambar menjadi lebih kabur.

Anti-aliasing dalam pengolahan sinyal digital adalah teknik mengurangi artifak distorsi dalam merepresentasikan citra resolusi tinggi pada resolusi yang lebih rendah. Artifak distorsi disebut aliasing. Anti-aliasing digunakan dalam fotografi digital, grafik komputer, audio digital, dan bidang lainnya.

Anti-aliasing berarti menghilangkan komponen sinyal yang memiliki frekuensi lebih tinggi dari yang dapat diterima oleh alat perekam (*sampling*). Jika perekaman dilakukan tanpa menghilangkan bagian sinyal ini, maka dapat menyepertemuankan tampilan citra yang tidak diinginkan (*noise*).

Line Thinning atau yang biasa disebut Anti Aliasing adalah teknik mengurangi jagged (tepi objek yang bergerigi) pada objek, agar dapat menampilkan resolusi tinggi pada resolusi yang lebih rendah, bisa juga Anti aliasing ini digunakan untuk mengurangi efek aliasing seperti gambar yang diperbesar akan membentuk sebuah tangga/garis patah-patah. Anti aliasing sejauh ini dibagi menjadi 2x,4x,8x,16x semakin tinggi tingkat aliasing nya maka semakin halus atau semakin menarik gambar/objek tersebut.

Interpolasi: Menciptakan Sesuatu Dari (Hampir) Tidak Ada?

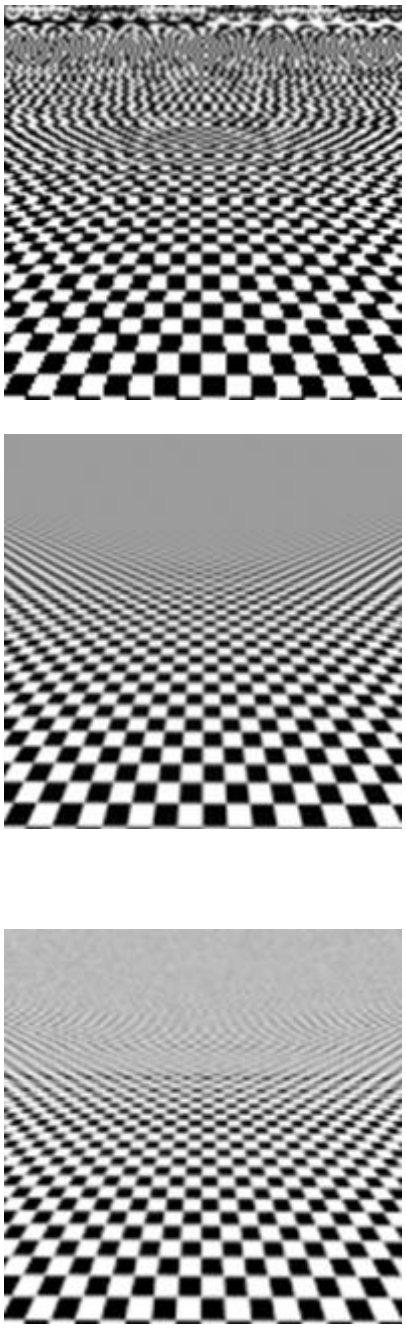


Gambar 11. 4 Interpolasi

Fotografi digital adalah perkiraan warna dan nilai yang ada ketika cahaya menimpa sensor-dengan cara yang sama, anti-aliasing adalah perkiraan data gambar menggunakan teknik yang disebut "Interpolasi." Interpolasi adalah istilah

istilah yang dibuat berdasarkan tren data yang ada, yaitu tebakan berpendidikan tentang apa yang sebenarnya ada di tempat itu jika lebih banyak titik data tersedia.

Meskipun lebih rumit daripada menebak-nebak, ada rumus dan metode yang tepat untuk Interpolasi, tidak dapat diharapkan untuk menjadi representasi akurat dari data gambar yang sebenarnya ada. Bahkan matematika paling pintar pun tidak bisa menciptakan sesuatu dari ketiadaan.



Gambar 11. 5 Checkerboards

Ketika kita melihat komputer ini membuat checkerboards, kita dapat mulai memahami apa yang dilakukan anti-aliasing untuk meningkatkan dan memperkirakan gambar. Pada gambar paling kiri, tidak ada interpolasi data – dam ditampilkan dalam piksel hitam dan putih saat kembali ke perspektif, dan dengan cepat menjadi berantakan. Kesalahan visual dan artefak yang dibuat adalah apa yang kita sebut “aliasing.” Gambar kedua dan ketiga di atas menggunakan berbagai bentuk “anti-aliasing” untuk lebih mendekati bagaimana mata manusia (dan kamera) merasakan cahaya.

Gambar 11. 6 Editing Photosop Alias

Gambar-gambar itu, bagaimanapun, adalah terjemahan dari gambar matematika absolut ke gambar berbasis piksel. Bagaimana anti-aliasing berlaku untuk fotografi Anda? Ketika gambar diubah ukurannya, diperbesar atau diperkecil, gambar diinterpolasi berdasarkan data yang ada dalam dokumen gambar. Gambar kiri menyusut menggunakan "tetangga terdekat" resampling di Photoshop-dengan kata lain, itu tidak anti-alias (Anda benar-benar dapat menyebutnya alias). Gambar di sebelah kanan dikurangi dan anti-alias, menciptakan gambar yang lebih benar pada ukuran yang kecil.

Gambar 11. 7 Memperbesar Gambar

Gambar yang diperbesar juga mendapat manfaat dari program anti-aliasing-grafis membuat tebakan terbaik mereka berdasarkan data di gambar Anda. Ingatlah ketika Anda memperbesar gambar-gambar dalam program grafis, bahwa Anda tidak akan pernah benar-benar mendapatkan resolusi lebih dari pembesaran digital-jenis interpolasi yang dilakukan dapat membuat tebakan yang baik tentang apa yang harus ada di sana, tetapi tidak akan pernah tahu pasti. Tepian Anda akan lembut, dan menjadi lebih lembut karena foto semakin membesar.

Aturan praktis yang baik adalah Anda selalu dapat menurunkan (mengecilkan) gambar Anda tanpa kehilangan kualitas dari anti-aliasing. Upsampling (memperbesar) membuat anti-aliasing sangat jelas, tidak menambahkan resolusi baru, dan seharusnya hanya dilakukan jika tidak dapat dihindari.

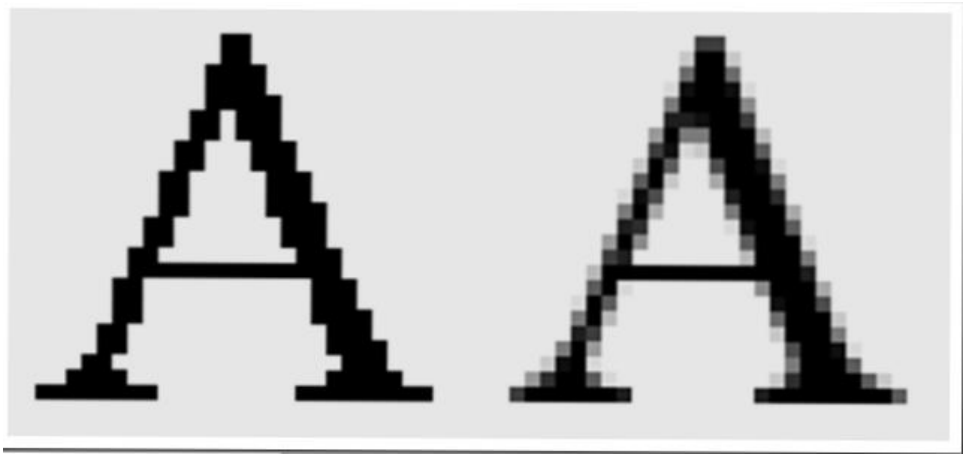
Anti-Aliasing dan Vektor: Mengapa Anti-Aliasing Membuat Videogame Terlihat Lebih Baik?

Gambar 11. 8 Video Gaming

Jika Anda telah memainkan game PC dalam 15 tahun terakhir atau lebih, Anda mungkin telah melihat opsi video yang menyertakan pengaturan untuk anti-aliasing. Jika Anda ingat 135olygo kami membahas bentuk 135olygo yang ada dalam posisi absolut, Anda harus mulai memahami mengapa anti-aliasing penting untuk permainan video.

Bentuk Dimensi dibuat dalam polygon-poligon, dan poligon-poligon ini ada di alam matematika saja.Anti-aliasing di video game memiliki setidaknya dua tujuan: pertama-tama ia ingin dapat membuat garis poligon mutlak dan beraturan dalam bentuk yang terlihat layak pada monitor berbasis piksel; kedua, anti-aliasing lebih baik mereplikasi dengan cara yang tidak tepat bahwa fotografi dan mata manusia merasakan cahaya.

Anti-Aliasing dan Tipografi



Gambar 11. 9 Tipografi Piksel dan Alias

Pada catatan terakhir, ada banyak kesempatan di mana anti-aliasing tidak ideal. Jika Anda pernah bekerja di sekitar desainer grafis, Anda mungkin pernah

mendengar mereka mengeluh tentang tipografi di Photoshop, dan betapa inferiornya terhadap Illustrator-dan mereka benar.

Kedua huruf di atas adalah tipografi berbasis piksel, dengan yang kiri adalah alias, yang benar anti-aliased. Bukan merupakan representasi tipografi yang baik, atau setidaknya tipografi itu. Hal ini dapat diterima untuk membuat font di layar dengan anti-aliasing, tetapi untuk mencetak, itu dapat memiliki beberapa konsekuensi bencana.

Gambar 11. 10 Suatu Logo Brand

Ketika Anda berpikir tentang huruf apa, mereka tidak benar-benar mengikuti aturan yang sama yang diperlukan fotografi digital. Huruf adalah ide abstrak dan bentuk mutlak – mereka lebih baik menjadi kategori “murni matematika” karya seni vektor. Dan tergantung pada jenis proses pencetakan yang digunakan untuk membuatnya, bentuk vektor matematika murni itu menjadi sangat penting.

Gambar 11. 11 Tipe Gambar Anti-aliased

Gambar ini di atas dibuat dengan tipe anti-aliased, dan kemungkinan besar offset dicetak. Ketika kita melihat lebih dekat kita dapat melihat mengapa itu buruk.

Gambar 11. 12 Pencitraan berbasis piksel

Jelas sangat cepat bahwa bentuk-bentuk anti-aliased tidak bertahan dengan baik ketika dicetak dengan cara ini. Ini adalah contoh bagaimana anti-aliasing (serta pencitraan berbasis piksel) dapat menjadi lebih rendah saat menampilkan tipografi.

Tentu saja, jika ini adalah gambar (seperti foto) dan bukan bentuk abstrak dari jenisnya, itu akan bertahan cukup baik.

Gambar 11. 13 Gambar menjadi media abstrak

Ketika, menjadi media abstrak, membutuhkan ketepatan vektor untuk bertahan di bawah jenis proses pencetakan yang tidak menggunakan titik-titik inkjet untuk membuat gambar. Bahkan pada jarak yang sangat dekat, kita tidak melihat titik atau bukti apa pun yang anti-aliasing yang masuk ke file yang digunakan untuk mencetak kaleng Coke ini.

Tentu saja, sebagian besar pembaca HTG tidak akan mengimbangi sebagian besar foto mereka, jadi tipografi berbasis piksel yang dicetak dari printer berbasis-titik akan bekerja dengan baik. Sadarilah anti-aliasing Anda saat Anda bekerja dengan tipografi dan ketika Anda bekerja dengan fotografi-Anda

akan menemukan Anda lebih siap untuk membuat pilihan yang tepat yang akan memberi Anda gambar terbaik.

Perlukah Penggunaan Anti Aliasing PC?

Perlu. karena game PC biasanya dimainkan pada layar 15 Inch keatas untuk grafis yang halus. Penggunaan Anti-Aliasing ini sangat menggunakan Kekuatan Hardwaremu, Pastikan Komponen PC mu kuat untuk digunakan Anti-Aliasing dan Cooling yang bagus agar tidak Overheat pada saat GPU bekerja keras.

Perlukah Pengunaan Anti-Aliasing di HP (Mobile)?

Untuk HP yang notabene tidak punya Cooler, Anti-Aliasing ini direkomendasikan Untuk di Matikan saja. Walaupun mayoritas Anti-aliasing yang digunakan adalah FXAA (Seperti PUBG Mobile) dan beberapa game yang punya SSAA (Resolution Scale) Sebaiknya disetarakan dengan Resolusi layar. Karena Jika Anti-Aliasing akan memaksa Hardware HP bekerja keras dan akhirnya menjadi Panas. Jika sudah seperti itu maka akan terjadi Thermal Throttling dimana HP ahrus menurunkan perfomanya agar tidak overhating. jika itu terjadi Pengalaman gaming mu bisa Pudar karena Frame rate yang turun.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 11	1. Jelaskan beberapa jenis antialiasing! 2. Jelaskan mengapa antialiasing membuat grafik lebih baik?

D. REFERENSI

PERTTEMUAN 12

GRAFIKA 3 DIMENSI

A. TUJUAN PEMAHAMAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu menerapkan grafika 3 dimensi.

Pada pertemuan ini akan dijelaskan mengenai :

1. Konsep Grafika 3D
2. Berbagai Proyeksi Dalam Grafika 3D

B. URAIAN MATERI

1. Konsep Grafika 3D

a. Grafika Komputer dalam aplikasinya terbagi menjadi 2 :

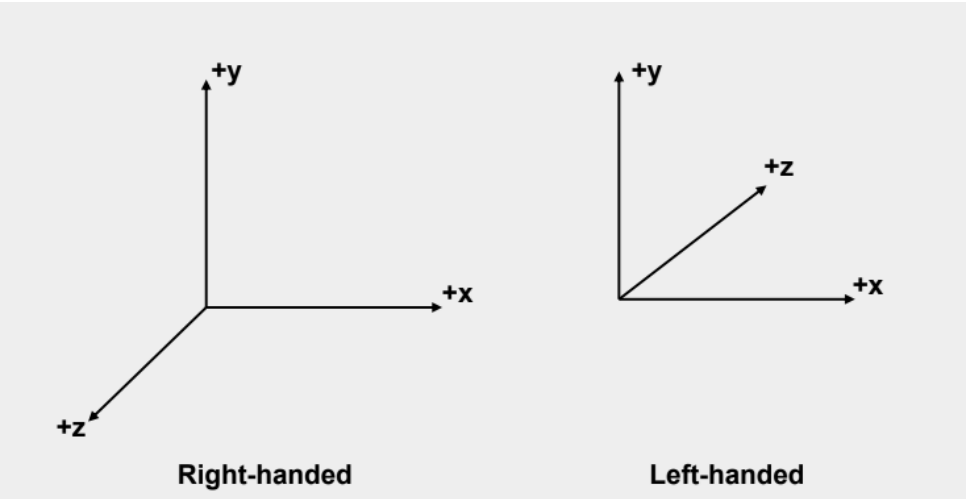
1) Grafika 2D

Aplikasi 2D banyak dipakai dalam pembuatan grafik, peta, kreasi 2D yang banyak membantu pemakai dalam membuat visualisasi. Grafika 2D memiliki kekurangan, yaitu : ketidakmampuannya untuk merepresentasikan objek 3D. Kekurangan ini sangat dirasakan terutama dalam bidang desain, dimana kebanyakan desainer membuat barang yang ada dalam dunia nyata yang berdimensi 3.

2) Grafika 3D

Grafika 3D memiliki kemampuan untuk membuat visualisasi dari sebuah benda iyang nyata yang dapat dilihat dari berbagai sudut pandang. Hal inilah yang membuat grafika 3D banyak dipakai terutama dalam bidang desain dari sebuah produk.

System Koordinat 3D



Gambar 12. 1 System Koordinat 3D

Dalam dunia 3D terdapat beberapa primitif seperti :

a) Titik (*point*)

Posisi sebuah titik dalam grafika 3D diekspresikan dengan (x,y,z)

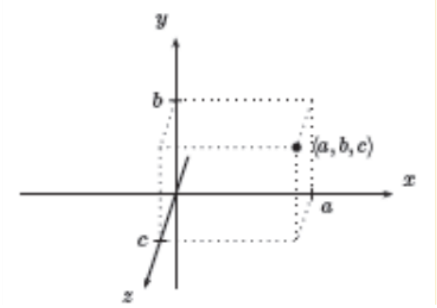
b) Garis (*line*)

(1) Sebuah garis dibentuk dengan mendeskripsikan dua buah titik, yaitu (x1,y1,z1) dan (x2,y2,z2) yang sebagai ujung dari sebuah garis.

(2) Sebuah garis dalam grafika 3D dapat diekspresikan dengan sepasang persamaan, yaitu :

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1}$$



c) Bidang/Permukaan (*planel/surface*)

Pada grafika 3D, terdapat sebuah geometri yang sangat penting, yaitu bidang datar (plane). Sebuah bidang datar pada grafika 3D dispesifikasikan dengan sebuah persamaan, yaitu :

$$Ax + By + Cz + D = 0$$

2. Berbagai Proyeksi Dalam Grafika 3D

Untuk merepresentasikan object 3D :

a. Persamaan Geometri

b. Constructive Solid Geometry (CSG)

c. Kurva & Permukaan Bezier

d. Lathe Object

e. Fractal

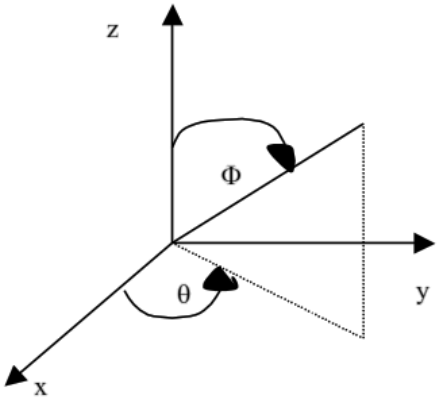
Dengan Persamaan Geometri:

1) Suatu object 3D dapat direpresentasikan langsung dengan menggunakan persamaan geometri dari object tersebut.

2) Misalkan : untuk membangun sebuah bola, maka bisa dengan menggunakan rumus :

$$X^2 + Y^2 + Z^2 = R^2$$

atau dengan rumus :



- $x = r.\sin\Phi.\sin\theta \quad ;0 \le \Phi \le 2\pi$
- $y = r.\sin\Phi.\cos\theta \quad ;-\pi \le \theta \le \pi$
- $z = r.\cos\Phi$

Dengan Constructive Solid Geometry (CSG):

a. CSG adalah suatu cara membentuk object dengan jalan menggabungkan atau memotong (mengurangi) dari beberapa object primitif 3D.

b. CSG dalam POV-Ray melibatkan :

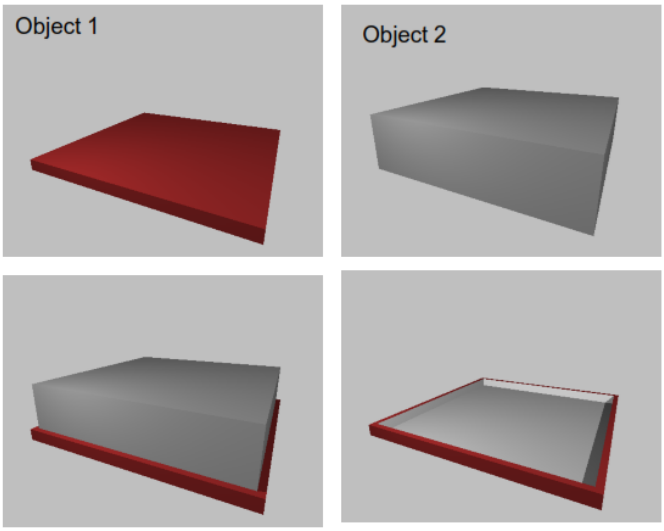
   a) Difference

   b) Union

   c) Intersect

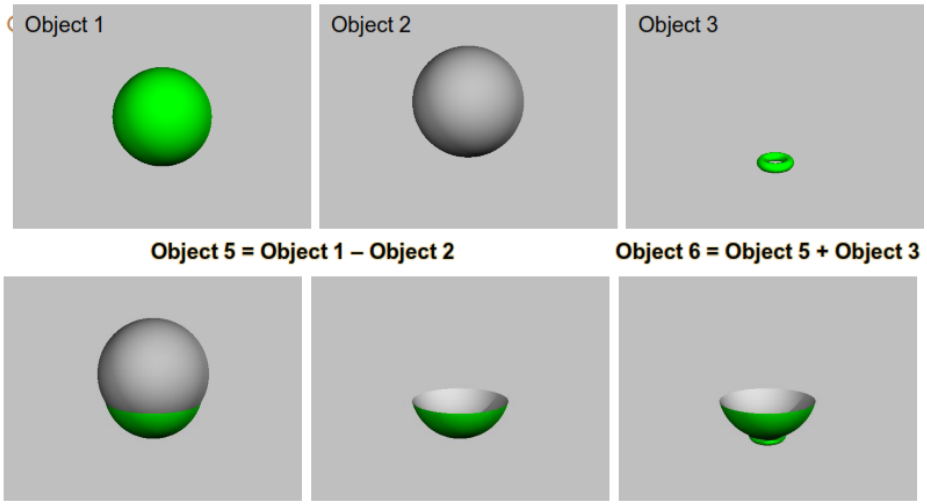
Dengan CSG – contoh

- Contoh 1:

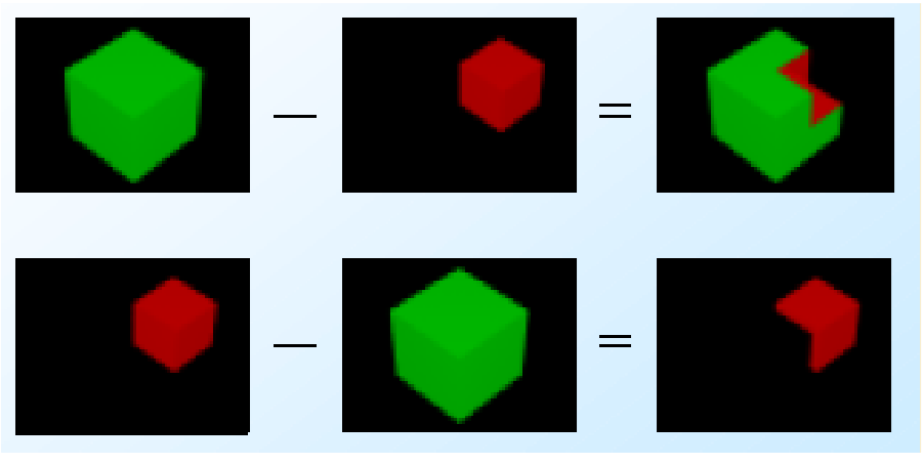


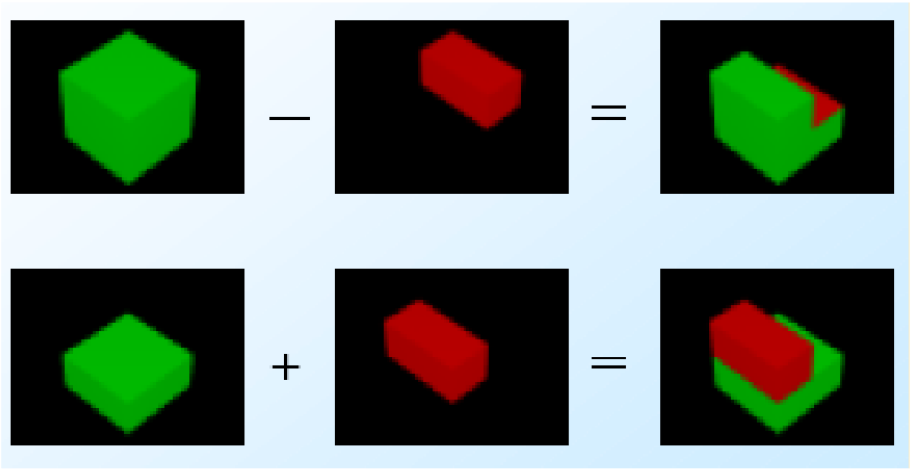
Gambar 12. 2 CSG

Dengan CSG – contoh



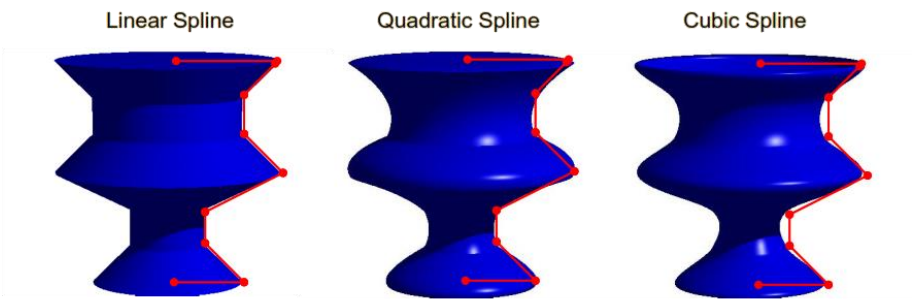
Gambar 12. 3 CSG 3 Object





Gambar 12. 4 CGS Pengurangan

Dengan Lathe Object



Gambar 12. 5 Lathe Object

Lathe merupakan salah satu perintah di 3ds Max yang digunakan untuk membentuk objek 2 dimensi menjadi 3 dimensi dengan cara diputar, dimana fitur ini sama seperti perintah revolve yang terdapat pada AutoCAD.

Perintah lathe ini terdapat pada tab modifier dan ada berbagai macam settingan untuk mempermudah anda membentuk objek 3 dimensi berdasarkan UCS atau sumbu x, y dan z.

a. Rendering

1) **Rendering** adalah proses akhir dari keseluruhan proses pemodelan ataupun animasi transparent.

2) Dalam rendering, semua data-data yang sudah dimasukkan dalam proses modeling, animasi, texturing, pencahayaan dengan parameter tertentu akan diterjemahkan dalam sebuah bentuk output (tampilan akhir pada model dan animasi)

3) Rendering tidak hanya digunakan pada game programming, tetapi juga digunakan pada banyak bidang, misalnya arsitektur, simulator, movie, transparent effect pada tayangan televisi, dan design visualization.

4) Rendering pada bidang-bidang tersebut memiliki perbedaan, terutama pada fitur dan transparan renderingnya. Terkadang rendering juga diintegrasikan dengan model yang lebih besar seperti paket animasi, tetapi terkadang berdiri sendiri dan juga bisa free open-source product.

5) Rendering ➔ Proses untuk menghasilkan sebuah citra 2D dari data 3D.

6) Prose ini bertujuan untuk untuk memberikan visualisasipada user mengenai data 3D tersebut melalui monitor atau pencetak yang hanya dapat menampilkan data 2D

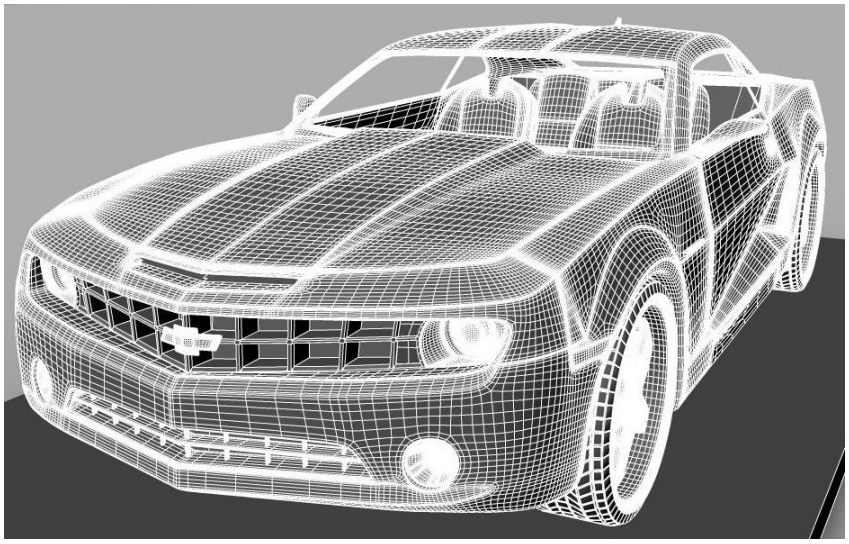
7) Metode rendering yang paling sederhana dalam grafika 3D :

a) Wireframe Rendering

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 transparent dengan sangat cepat, hanya kelemahannya adalah tidak adanya permukaan, sehingga sebuah objek terlihat transparent. Sehingga sering terjadi kesalahpahaman antara sisi depan dan sisi belakang dari sebuah objek.



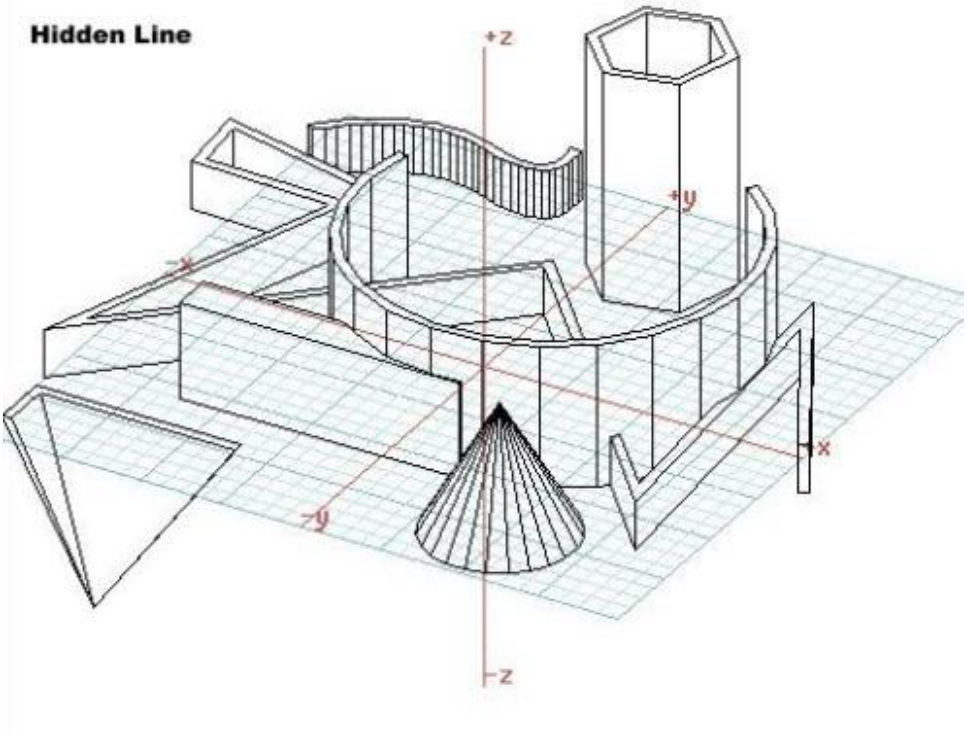
Gambar 12. 6 Wireframe Rendering

b) 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 igaris-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.

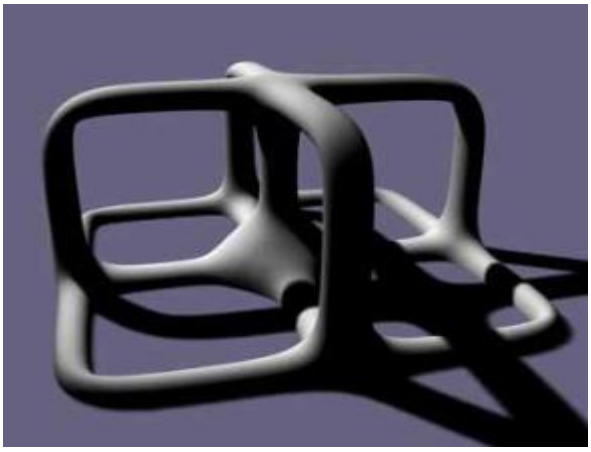


Gambar 12. 7 Line Rendering

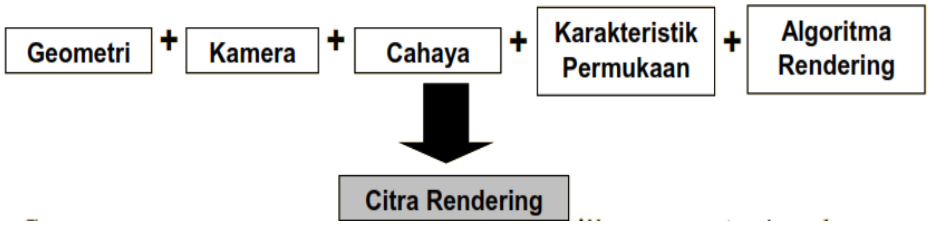
c) Shaded Rendering

Pada metode ini, komputer diharuskan untuk melakukan berbagai perhitungan baik pencahayaan, karakteristik permukaan, shadow casting, dan lain-lain

Metode ini menghasilkan citra yang sangat realistik, tetapi kelemahannya adalah lama waktu rendering yang dibutuhkan.



12. 8 Shaded Rendering



Gambar 12. 9 Proses Rendering

Secara umum, proses untuk menghasilkan rendering dua dimensi dari objek-objek 3D melibatkan 5 komponen utama :

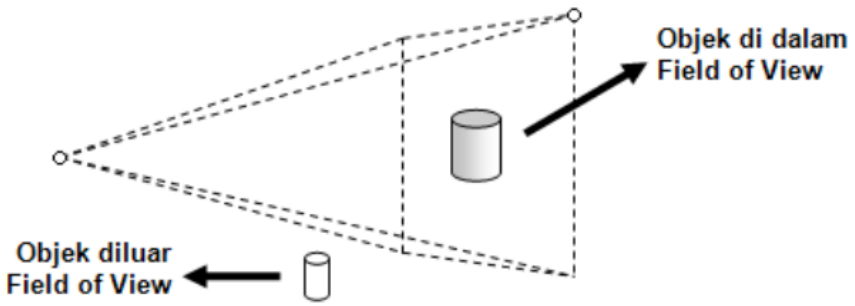
3) Geometri
4) Kamera
5) Cahaya
6) Karakteristik Permukaan
7) Algoritma Rendering

b. Kamera

Dalam grafika 3D, sudut pandang (*point of view*) adalah bagian dari kamera. Kamera dalam grafika 3D biasanya tidak didefinisikan secara fisik, namun hanya untuk menentukan sudut pandang kita pada sebuah world, sehingga sering disebut virtual camera.

Sebuah kamera dipengaruhi oleh dua buah faktor penting.

1) Faktor pertama adalah lokasi *(camera location*). Lokasi sebuah kamera ditentukan dengan sebuah titik (x,y,z).
2) Faktor kedua adalah arah pandang kamera. Arah pandang kamera ditunjukkan dengan sebuah sistem yang disebut sistem koordinat acuan pandang atau sistem (U,N,V). Arah pandang kamera sangat penting dalam membuat sebuah citra, karena letak dan arah pandang kamera menentukan apa yang terlihat oleh sebuah kamera. Penentuan apa yang dilihat oleh kamera biasanya ditentukan dengan sebuah titik (x,y,z) yang disebut camera interest.



Gambar 12. 10 Point of View

Pada kamera, dikenal ***field of view*** yaitu daerah yang terlihat oleh sebuah kamera.Field of view pada grafika 3D berbentuk piramid, karena layar monitor sebuah komputer berbentuk segiempat. Objek-objek yang berada dalam field of view ini akan terlihat dari layar monitor, sedang objek-objek yang berada di luar *field of view* iini tidak terlihat pada layar monitor. *Field of view* ini sangat penting dalam pemilihan objek yang akan diproses dalam rendering. Objek- objek diluar *field of view* biasanya tidak akan diperhitungkan, sehingga perhitungan dalam proses rendering, tidak perlu dilakukan pada seluruh objek.

Sumber cahaya pada grafika 3D merupakan sebuah objek yang penting, karena dengan cahaya ini sebuah world dapat terlihat dan dapat dilakukan proses rendering. Sumber cahaya ini juga membuat sebuah world menjadi lebih realistis dengan adanya bayangan dari objek- objek 3D yang ada.

sumber cahaya memiliki jenis. Pada grafika 3D dikenal beberapa macam sumber cahaya, yaitu :

1) Point Light

   Memancar ke segala arah, namun intensitas cahaya yang diterima objek bergantung dari posisi sumber cahaya. Tipe ini mirip seperti lampu pijar dalam dunia nyata.

2) Spotlight

   Memancarkan cahaya ke daerah tertentu dalam bentuk kerucut. Sumber icahaya terletak pada puncak kerucut. Hanya objek-objek yang terletak pada daerah kerucut tersebut yang akan nampak.

3) Ambient Light

Cahaya latar/alam. Cahaya ini diterima dengan intensitas yang sama oleh isetiap permukaan pada benda. Cahaya latar tersebut dimodelkan mengikuti apa yang terjadi di alam, dalam keadaan tanpa sumber cahaya sekalipun, benda masih dapat dilihat.

4) Area Light

5) Directional Light

Memancarkan cahaya dengan intensitas sama ke suatu arah tertentu. iLetak tidak mempengaruhi intensitas cahayanya. Tipe ini dapat menimbulkan efek seolah-olah sumber cahaya berada sangat jauh dari objek.

6) Parallel Point

Sama dengan directional, hanya pencahayaan ini memiliki arah dan posisi.

Model dari pencahayaan, dipakai untuk menghitung intensitas dari cahaya yang terlihat dari setiap posisi pada setiap permukaan benda yang terlihat oleh kamera. Ketika melihat sebuah benda, terlihat cahaya yang dipantulkan dari permukaan benda, dimana icahaya ini merupakan intregrasi dari sumber-sumber cahaya serta cahaya yang berasal dari pantulan cahaya permukaan-permukaan yang lain. Karena itu benda-benda yang tidak langsung menerima cahaya dari sumber cahaya, masih mungkin terlihat bila menerima cahaya pantulan yang cukup dari benda didekatnya.

Model sederhana dari sumber cahaya adalah sebuah titik sumber, dimana dari titik ini icahaya dipancarkan. Perhitungan pencahayaan bergantung pada sifat dari permukaan yang terkena cahaya, kondisi dari cahaya latar serta spesifikasi sumber cahaya.

Semua **sumber cahaya dimodelkan sebagai sumber titik** yang dispesifikasikan dengan :

1) **Lokasi** ➔ Lokasi (x,y,z) dari sebuah sumber cahaya akan menentukan pengaruhnya terhadap sebuah objek.

2) **Intensitas** ➔ Intensitas cahaya menyatakan kekuatan cahaya yang dipancarkan oleh sebuah sumber cahaya. Parameter ini merupakan

angka, yang biasanya makin besar nilainya, makin terang sumber cahaya tersebut.

3) **Warna** ➔ Warna cahaya dari sumber ini akan mempengaruhi warna dari sebuah objek, jadi selain warna objek tersebut warna cahaya yang jatuh pada objek tersebut akan mempengaruhi warna pada rendering. Warna cahaya ini biasanya terdiri dari 3 warna dasar grafika komputer, yaitu: merah, hijau, biru atau lebih dikenal dengan RGB.

c. Karakteristik Permukaan

Karakteristik permukaan dari sebuah objek adalah sifat dari permukaan sebuah objek. Karakteristik permukaan ini meliputi:

1) Warna
2) Tekstur
3) Sifat permukaan, seperti kekasaran (roughness), refleksifitas, diffuseness (jumlah cahaya yang dipantulkan oleh objek), transparansi, dan lain-lain.

Parameter **Warna** dalam karakteristik permukaan direpresentasikan dengan tiga iwarna dasar, yaitu RGB. Saat rendering, warna pada sebuah objek tergantung dari warna dalam karakteristik permukaan dan warna cahaya yang mengenainya. Jadi citra hasil rendering mungkin akan memiliki warna yang sedikit berbeda dengan warna objek tersebut.

Parameter **Tekstur** direpresentasikan dengan sebuah nama file. File ini akan menjadi tekstur pada permukaan objek tersebut. Selain itu juga ada beberapa iparameter dalam tekstur yang berguna untuk menentukan letak tekstur pada sebuah objek, sifat tekstur, perulangan tekstur, dan lain-lain.

Sifat Permukaan, seperti diffuseness, refleksisifitas, dan lain- lain direpresentasikan dengan sebuah nilai. Nilai ini menentukan sifat dari parameter-parameter tersebut. Misalnya pada roughness, makin besar nilai parameternya, makin kasar objek tersebut.

d. Algoritma Rendering

Algoritma Rendering adalah prosedur yang digunakan oleh suatu program untuk mengerjakan perhitungan untuk menghasilkan citra 2D dari data 3D.

algoritma rendering yang ada saat ini menggunakan pendekatan yang disebut ***scan-line rendering*** ➔ berarti program melihat dari setiap pixel, satu per satu, secara horizontal dan menghitung warna di pixel tersebut.

Saat ini dikenal 3 algoritma :

1) *Ray-Casting*
2) *Ray-Tracing*
3) *Radiosity*

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 12	1. Sebutkan beberapa primitif Grafik 3D ? 2. Jelaskan fungsi dari Rendering ? 3. Jelaskan & Sebutkan beberapa macam sumber cahaya pada Grafika 3D ?

D. REFERENSI

PERTEMUAN 13

PEMROGRAMAN KOMPUTER GRAFIK MENGUNAKAN API

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu merancang pemrograman komputer grafik menggunakan API.

Pada pertemuan ini akan dijelaskan mengenai :

1. OpenGL
2. Pengenalan Pengolahan Citra Digital

B. URAIAN MATERI

1. Pengertian OpenGL

OpenGL awalnya disebut **Open Graphics Library** adalah Antarmuka *pemrograman* aplikasi (API) grafik 2D dan 3D yang dikembangkan oleh Silicon Graphics Inc. (SGI) untuk workstation Unix-nya. Meskipun versi OpenGL SGI telah ada sejak lama, spesifikasi OpenGL standar pertama muncul pada tahun 1992. Sekarang diadopsi secara luas di semua sistem operasi, standar OpenGL membentuk dasar dari sebagian besar permainan, desain berbantuan komputer (CAD) , dan bahkan industri virtual reality (VR).

OpenGl dirancang sebagai antarmuka yang efisien dan tidak bergantung pada perangkat keras untuk di terapkan pada banyaknya platform perakat kelas untuk diterapkan pada banyak platform perangkat keras yang berbeda. Untuk menciptakan kulitas ini, tidak ada perintah untuk melakukan tugas windowing atau mendapatkan input pengguna yang disertakan dalam OpenGL. Sebaliknya, anda harus bekerja melalui system windowing apa pun yang mengontrol perangkat keras tertentu yang anda gunakan. Demikian pula, OpenGL pula, OpenGL tidak menyediakan perintah tingkat tinggi untuk mendeskripsikan modal objek 3D. Perintah semacam itu memungkinkan anda untuk menentukan bentuk relative rumit seperti mobil,bagian tubuh, pesawat terbang, atau molekul.

Dengan OpenGl, anda harus membangun modal yang diinginkan dari sekumpulan kecil geometris primitive, titik, garis dan poligon

API adalah singkatan dari antarmuka pemrograman aplikasi (API) adalah antarmuka komputasi yang mendefinisikan interaksi antara beberapa perantara perangkat lunak. Ini mendefinisikan jenis panggilan atau permintaan yang dapat dibuat, bagaimana membuatnya, format data yang harus digunakan, konvensi yang harus diikuti, dll. Ia juga dapat menyediakan mekanisme ekstensi sehingga pengguna dapat memperluas fungsionalitas yang ada dengan berbagai cara dan untuk berbagai tingkat. API dapat sepenuhnya dibuat khusus, spesifik untuk suatu komponen, atau dirancang berdasarkan standar industri untuk memastikan interoperabilitas. Melalui penyembunyian informasi, API mengaktifkan pemrograman modular, memungkinkan pengguna menggunakan antarmuka secara independen dari implementasi. Karena OpenGL adalah API grafik dan bukan platformnya sendiri. Fungsi dasar dari OpenGL adalah untuk mengeluarkan koleksi perintah khusus atau executable ke sistem operasi. Dengan demikian, program ini bekerja dengan perangkat keras grafis yang ada yang berada pada hard drive atau sumber tertentu lainnya. Setiap perintah dalam dirancang untuk melakukan tindakan tertentu, atau memulai efek khusus tertentu yang terkait dengan grafis.

a. Langkah-langkah OpenGl
   1) Install Microsoft Visual Studio.NET pada komputer anda.
   2) Siapkan file OpenGL95.exe dan glut-3.7.6.zip pada direktori sementara
   3) Masukan :
      a) GL.H, GLAUX.H, GLU.H dan glut.h ke drive:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\include\gl
      b) GLAUX32.LIB, GLU32.LIB, OPENGL32.LIB dan glut32.lib ke drive:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\lib
      c) OPENGL32.DLL, GLU32.DLL dan glut32.dll ke drive:\Windows\System
b. Membuat Program OpenGL Sederhana
   1) Jalankan Visual Studio .NET, buatlah sebuah proyek dengan tipe Visual C++ Projects dan template Win32 Console Project. Beri nama sesuai selera anda.
   2) Pada Application Settings pilih Console Application dan Empty Project

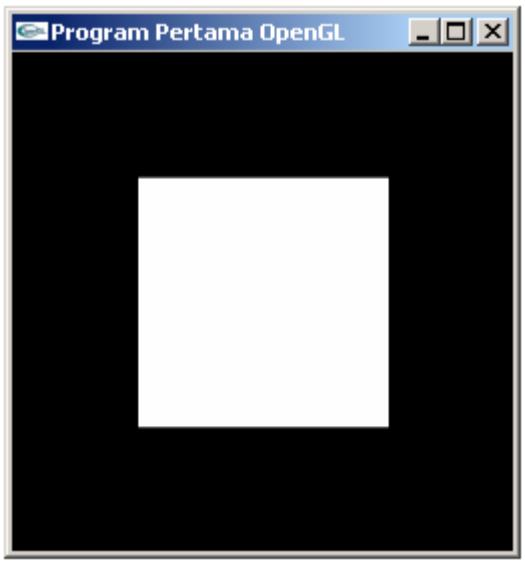
3) Pada Solution Explorer, klik kanan Source Files lalu Add - Add New Item,lalu pilih template C++ File (.cpp), beri nama menurut selera anda, klik Open. Pada layar akan terlihat halaman kosong
4) Masukkan program yang berikut ini:

```
#include
#include void display(void)
{
glClear (GL_COLOR_BUFFER_BIT);
glColor3f (1.0, 1.0, 1.0);
glBegin(GL_POLYGON);
glVertex3f (0.25, 0.25, 0.0);
glVertex3f (0.75, 0.25, 0.0);
glVertex3f (0.75, 0.75, 0.0);
glVertex3f (0.25, 0.75, 0.0);
glEnd();

}
int main(int argc, char** argv)
{ glutInit(&argc, argv);
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize (250, 250);
glutInitWindowPosition (100, 100);
glutCreateWindow ("Program Pertama OpenGL");
glClearColor (0.0, 0.0, 0.0, 0.0);
175 glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0); glutDisplayFunc(display);
glutMainLoop();
return 0;
}
```

5) Modifikasi Project Properties dengan memilih Project – Properties – All Configurations – Linker, masukkan opengl32.lib glu32.lib glut32.lib pada textbox Additional Dependencies f.

6) Jika diperlukan, agar console windows tidak dibuka pada saat menjalankan program (console output akan disabled), masukan /SUBSYSTEM:WINDOWS /ENTRY:mainCRTStartup pada Linker – Command Line – Additional Options g.
7) Lakukan kompilasi terhadap program tersebut, hasilnya adalah sebagai berikut



Gambar 13. 1 Program Pertama OpenGL

2. Pengenalan Pemrosesan Citra Digital

a. Pengertian Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog

adalah citra yang bersifat *continue* seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer. Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan

(*grayscale*) sesuai dengan jurnlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.

b. Jenis – jenis Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara ipenyimpanan menentukan jenis citra digital yang terbentuk. Beberapa jenis citra digital yang sering digunakan yaitu :

1) Citra Biner (Monokrom)

Banyaknya dua warna, yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini.

2) Citra *Grayscale* (Skala Keabuan)

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk.

3) Citra Warna (*True Color*).

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RG8 = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak 28 x 28 x 28 = 224 =16 juta warna lebih. Itulah sepertemuannya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

c. Elemen- elemen Citra digital

Berikut adalah elemen-elemen yang terdapat pada citra digital yaitu sebagai berikut :

1) Kecerahan (*Brightness*).

Brightness merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2) Kontras (*Contrast*).

Kontras menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

3) Kontur (*Contour*).

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4) Warna *(Colour).*

Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek.

5) Bentuk (*Shape*).

Shape adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia.

6) Tekstur *(Texture).*

Texture dicirikan sebagai distribusi spasial dari derajat keabuan di dalam *isekumpulan* piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar, sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

d. Pengertian Pemrosesan Citra Digital

Pemrosesan atau pengolahan citra digital (*digital image processing*) merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra. Pemrosesan citra digital secara diagram pemrosesan dimulai dari pengambilan citra, perbaikan kualitas citra, sampai dengan pernyataan representatif citra yang dicitrakan sebagai berikut:

Gambar 13. 2 Diagram pemrosesan citradigital

Dalam perkembangan lebih lanjut, *Digital image processing* dan *computer vision* digunakan sebagai mata manusia, dengan perangkat *input image capture* seperti kamera dan *scanner* dijadikan sebagai mata dan mesin komputer (dengan program komputasinya) dijadikan sebagai otak yang mengolah informasi. Sehingga muncul beberapa pecahan bidang yang menjadi penting dalam *computer vision*, antara lain: *pattern recognition* (pengenalan pola), *biometric* pengenalan identifikasi manusia berdasarkan ciri-ciri biologis yang tampak pada badan manusia), *content based image and video retrieval* (mendapatkan kembali citra atau video dengan informasi tertentu), video editing, dan lain-lain

e. Teknik – teknik pengolahan citra digital

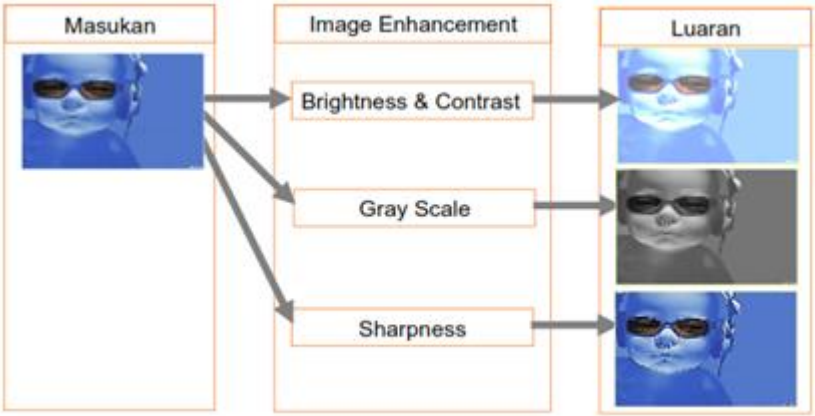
Secara umum, teknik pengolahan citra digital dibagi menjadi tiga tingkat pengolahan, yakni sebagai berikut:

1) Pengolahan Tingkat Rendah (*Low-Level Processing*). Pengolahan ini merupakan operasional-operasional dasar dalam pengolahan citra, seperti pengurangan noise (*noise reduction*), perbaikan citra (*image enhancement*) dan restorasi citra (*image restoration*).

2) Pengolahan Tingkat Menengah (*Mid-Level Processing*). Pengolahan ini meliputi segmentasi pada citra, deskripsi objek, dan klasifikasi objek secara terpisah.
3) Pengolahan Tingkat Tinggi (*High-Level Processing*). Pengolahan ini meliputi analisis Citra.

Dari ketiga tahap pengolahan citra digital di atas, dapat dinyatakan suatu gambaran mengenai teknik-teknik pengolahan citra digital dan macam-macamnya, antara lain sebagai berikut :

1) ***Image enhancement***, berupa proses perbaikan citra dengan meningkatkan kualitas citra, baik kontras maupun kecerahan.



Gambar 13. 3 Image enhancement

2) ***Image restoration***, yaitu proses memperbaiki model citra,biasanya berhubungan dengan bentuk citra yang sesuai.

Gambar 13. 4 Image restoration

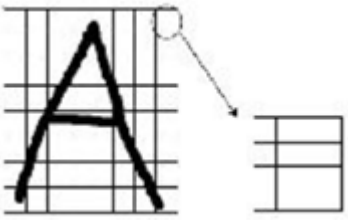
3) ***Color image processing***, yaitu suatu proses yang melibatkan citra berwarna, baik berupa *image enhancement*, *image restoration*, atau yang lainnya.

Gambar 13. 5 Color image processing

4) ***Wavelet dan multiresolution processing***, merupakan suatu proses yang menyatakan citra dalam beberapa resolusi.

Gambar 13. 6 Wavelet dan multiresolution processing

5) ***Mage compression*****,** merupakan proses yang digunakan untuk mengubah ukuran data pada citra.



Gambar 13. 7 Mage compression

6) ***Morphological processing***, yaitu proses untuk memperoleh informasi yang menyatakan deskripsi dari suatu bentuk pada citra.

Gambar 13. 8 Morphological processing

7) ***Segmentation*****,** merupakan proses untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, seperti memisahkan objek dengan latar belakangnya.

Gambar 13. 9 Segmentation

8) ***Object recognition*****,** yaitu suatu proses yang dilakukan untuk mengenali objek-objek apa saja yang ada dalam suatu citra.

Gambar 13. 10 Object recognition

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 13	1. Carilah aplikasi pengolah citra di internet, kemudian analisalah aplikasi tersebut berdasarkan teori diatas!

D. REFERENSI

PERTEMUAN 14

APLIKASI GRAFIKA KOMPUTER

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu membuat aplikasi grafika computer.

B. URAIAN MATERI

1. Aplikasi Grafika komputer

Contoh Soal Membuat Aplikasi Grafika Komputer beserta pemograman nya, gambarkan animasi menggunakan Pemograman Java2d

Tambahkan lah menu label UAS .

Untuk source code nya, silahkan ketik source code berikut :

```
/*
* To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package UJIAN;


/**
 *
 * @author lenovo
 */
import java.awt.BasicStroke;
import java.awt.Color;
```

```
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.GeneralPath;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

/**
 *
 * @author lenovo
 */
public class animasi extends JPanel  {
    //Membuat variabel thread

    Thread animasi, repaint;
    int x=10,y=10,a=10, batas = 240;
    int xOrang = -100, yOrang = 100, v = 1;

    public animasi() {
        setPreferredSize(new Dimension(500, 300));
        setFocusable(true);
        setBackground(Color.black);
        requestFocusInWindow();
        initThread();
```

```
        //untuk memulai thread
        animasi.start();
        repaint.start();
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g;
        drawText(g2);
        drawOrang(g2, xOrang, yOrang, v);
    }

    //method untuk menampung thread
    public void initThread() {
        animasi = new Thread(new Runnable() {

            public void run() {
                while (true) {
                    //  <editor-fold  defaultstate="collapsed"  desc="menggerakkan
Orang">
                    if (xOrang < 500) {
                        if (v == 1) {
                            xOrang = xOrang + 10;
                            v = 2;
                        } else {
                            xOrang = xOrang + 10;
```

```
                    v = 1;
                }
            } else {
                xOrang = -100;
            }
            // </editor-fold>
            try {
                //untuk mengatur kecepatan thread, semakin kecil nilai semakin cepat
                Thread.sleep(150);
            } catch (Exception e) {
            }
        }
    }
});

repaint = new Thread(new Runnable() {

    public void run() {
        while (true) {
            // <editor-fold defaultstate="collapsed" desc="Untuk mengacak warna">
            if(a<batas){
                a=a+15;
                batas=240;
            }else{
                a=a-15;
```

```
                batas=15;
            }
            // </editor-fold>
            // <editor-fold defaultstate="collapsed" desc="menggerakkan
text">
            x=(int)(Math.random()*10);
            y=(int)(Math.random()*10);
            // </editor-fold>
            try {
                //untuk mengatur kecepatan thread, semakin kecil nilai
semakin cepat
                Thread.sleep(10);
            } catch (Exception e) {
            }
            //untuk memanggil repaint
            SwingUtilities.invokeLater(new Runnable() {

                public void run() {
                    repaint();
                }
            });
        }
    }
});
}

public void drawText(Graphics2D g2) {
```

```
            g2.setFont(new Font("Comic sans ms", 1, 35));
            g2.setColor(newColor((int)(Math.random()*250),
(int)(Math.random()*250), (int)(Math.random()*250), a));
            g2.drawString("DWI SUTIKNO", 20+x, 50+y);
        }
        public void drawOrang(Graphics2D g2, int x, int y, int v) {
            g2.setStroke(newBasicStroke(6,BasicStroke.CAP_ROUND,
BasicStroke.JOIN_ROUND));
            Ellipse2D kepala = new Ellipse2D.Double(x + 38, y + 10, 20, 20);
            GeneralPath gp = new GeneralPath();
            gp.moveTo(x + 46, y + 32);
            gp.lineTo(x + 34, y + 52);
            gp.lineTo(x + 27, y + 73);
            gp.moveTo(x + 46, y + 32);
            gp.lineTo(x + 54, y + 55);
            gp.lineTo(x + 71, y + 72);
            gp.moveTo(x + 46, y + 32);
            gp.lineTo(x + 46, y + 69);
            gp.lineTo(x + 46, y + 93);
            gp.lineTo(x + 26, y + 112);
            gp.moveTo(x + 46, y + 69);
            gp.lineTo(x + 59, y + 93);
            gp.lineTo(x + 66, y + 118);
      //=========================================================
======
            Ellipse2D kepala1 = new Ellipse2D.Double(x + 42, y + 10, 20, 20);
            GeneralPath gp1 = new GeneralPath();
            gp1.moveTo(x + 46, y + 29);
```

```
        gp1.lineTo(x + 40, y + 48);
        gp1.lineTo(x + 40, y + 77);
        gp1.moveTo(x + 46, y + 29);
        gp1.lineTo(x + 49, y + 58);
        gp1.lineTo(x + 51, y + 74);
        gp1.moveTo(x + 46, y + 29);
        gp1.lineTo(x + 46, y + 68);
        gp1.lineTo(x + 52, y + 90);
        gp1.lineTo(x + 46, y + 120);
        gp1.moveTo(x + 46, y + 68);
        gp1.lineTo(x + 60, y + 92);
        gp1.lineTo(x + 38, y + 106);

        g2.setPaint(Color.white);

        if (v == 1) {
            g2.fill(kepala);
            g2.draw(gp);
            g2.draw(kepala);
        } else if (v == 2) {
            g2.fill(kepala1);
            g2.draw(gp1);
            g2.draw(kepala1);
        }
    }

    public void xplay() {
```

```
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                final JFrame frame = new JFrame("Animasi Orang");
                frame.setDefaultCloseOperation(
                        JFrame.DISPOSE_ON_CLOSE);
                frame.add(new animasi());
                frame.pack();
                frame.setResizable(false);
                frame.setLocationRelativeTo(null);
                frame.setVisible(true);
            }
        });


    }
    // </editor-fold>
     public static void main(String salis[]) {
        new animasi().xplay();
    }
}
```

Berikut ouput nya

Gambar 14. 1 Output Animasi Orang

Untuk nomor 2 silahkan ketik source code berikut :

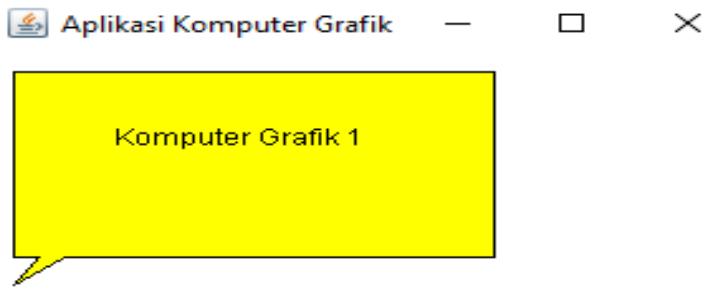
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package UJIAN;

import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
/**
 *
 * @author lenovo
 */
```

```
public class uas extends JPanel{
    public uas(){
        this.setPreferredSize(new Dimension(300,250));
        this.setBackground(Color.WHITE);
    }
    protected void paintComponent(Graphics g){
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g;
        Polygon kotak = new Polygon();
        kotak.addPoint(10,10);
        kotak.addPoint(200,10);
        kotak.addPoint(200,110);
        kotak.addPoint(30,110);
        kotak.addPoint(10,125);
        kotak.addPoint(20,110);
        kotak.addPoint(10,110);
        g2.setColor(Color.yellow);
        g2.fill(kotak);
        g2.setColor(Color.black);
        g2.drawString ("Komputer Grafik 1",50,50);
        g2.draw(kotak);
    }
    public static void main(String [] args){
        JFrame frame = new JFrame("Aplikasi Komputer Grafik");
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
        uas asik = new uas();
        frame.getContentPane();
        frame.add(asik);
        frame.pack();
        frame.setVisible(true);
    }
}
```

Berikut hasil ouput nya :



Gambar 14. 2 Output Aplikasi Komputer Grafik

Silahkan pelajari pertemuan 10 – 13

2. Bahan UAS

Buat tugas kelompok untuk membuat Source code aplikasi grafika komputer dengan Bahasa phyton.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 14	1. Buatlah Tulisan “Selamat Datang Tangerang“ di bingkai Bulat dengan Bahasa pemograman Ruby.

D. REFERENSI

GLOSARIUM

Komputer adalah mesin yang dapat diinstruksikan untuk menjalankan urutan aritmatika atau operasi logika secara otomatis melalui pemrograman komputer, komputer digunakan sebagai alat pengoprasian atau penggunaan mesin untuk melakukan peloses perhitungan.

Komputer grafik adalah ilmu dan seni berkomunikasi secara visual melalui tampilan komputer dan perangkat interaksinya

Pixel adalah unit terkecil dari gambar atau grafik digital yang dapat ditampilkan dan direpresentasikan pada perangkat tampilan digital

Resolusi adalah kehalusan detail dalam gambar bitmap dan diukur dalam piksel per inci (ppi).

Grafik komputer adalah bidang yang diminati banyak orang. Salah satu subbagian dari grafik komputer adalah pemodelan objek. Pada pemodelan objek dua dimensi (2D), ditemukan berbagai objek yang apat dimodelkan sesuai dengan kondisi tertentu, dan objek yang dimodelkan perlu dimodifikasi

Transformasi translasi adalah operasi yang menyepertemuankan objek 2D berpindah dari satu posisi ke posisi lain

Dilatasi adalah transformasi yang membuat mirror (pencerminan) dari image suatu objek.

DAFTAR PUSTAKA

Donald D. Hearn, M. Pauline, Warren Carithers, 2011, Computer Graphics with Open GL (4th Edition), Prentice-Hall,

Vince, John A., 2010,Mathematics for Computer Graphics (Undergraduate Topics for Computer Science). 3rd. Springer.

Hong Zang dan Y. Daniel Liang, 2006, Computer Graphics using Java 2D and 3D, Prentice Hall.

Hill, F.S., 2001, Computer Graphics Using OpenGL, Prentice Hall, Upper Saddle River, NJ.

ebook Irfan Maliki, S.T – Grafika Komputer ilmu Komputer

Dr. Setiawan Hadi - Modul Diktat Grafika Komputer

Pat Hanrahan, “Introductory Computer Graphics”, https://graphics.stanford.edu/courses/#cs148

-, “The Java Tutorials”, http://download.oracle.com/javase/tutorial/

-, ”Trail: 2D Graphics (The Java Tutorial)”, http://download.oracle.com/javase/tutorial/2d/index.html

-, “Java 3D API Tutorial”, http://java.sun.com/developer/onlineTraining/java3d/

RENCANA PEMBELAJARAN SEMESTER(RPS)

Program Studi : Teknik Informatika S-1

Semester : 6

Deskripsi Mata Kuliah : Mata kuliah ini merupakan mata kuliah wajib program studi Teknik Informatika S-1 dengan materi yang dibahas: Pengenalan Grafika Komputer, Titik dan Palet Warna, Algoritma Garis Dan Poligon, Algoritma Lingkaran dan Elips, Algoritma Pembuatan Kurva, Algoritma Pewarnaan Bidang, Proyeksi Geometri Bidang, Transformasi, Translasi dan Rotasi, Dilatasi, Viewing dan Clipping, Antialiasing, Grafika 3 Dimensi, Mampu Mengunakan Api Untuk Pemrograman Komputer Grafik, dan Membuat Aplikasi Grafika Komputer.

Penyusun :
1. Achmad Udin Zailani, S.Kom.,M.Kom
2. Hiram Sulistio Sibarani, S.Kom.,M.Kom
3. Agung Perdananto, S.Kom.,M.Kom

Mata Kuliah/Kode : Komputer Grafik 1/ TPL0362

Prasyarat : --

Sks : 2 Sks

Kurikulum : KKNI

Capaian Pembelajaran : Setelah menyelesaikan mata kuliah ini, mahasiswa mampu membuat aplikasi grafika komputer melalui kode program dengan baik.

PERTEMUAN KE-	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN (MATERI AJAR)	METODE PEMBELAJARAN	PENGALAMAN BELAJAR MAHASISWA	KRITERIA PENILAIAN	BOBOT NILAI
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	Mahasiswa mampu memahami pengenalan komputer grafik.	Pengenalan Komputer Grafik 1. Penjelasan metode danmateri perkuliahan 2. Pengenalan komputer grafik dan evolusinya 3. Pengenalan contoh pemanfaatan komputer grafik	Ceramah, Latihan, Diskusi	Mengajukan pertanyaan dan mengungkapkan tanggapan	Keaktifan mahasiswa dalam mengajukan pertanyaan dan mengungkapkan pertanyaan	5%
2	Mampu mengaplikasikan grafika komputer	Pengenalan Grafika Komputer 1. Membaca Dan Menulis Pixel 2. Membuat Palet Warna	Ceramah, Latihan, Diskusi	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	5%

PERTEMUAN KE-	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN (MATERI AJAR)	METODE PEMBELAJARAN	PENGALAMAN BELAJAR MAHASISWA	KRITERIA PENILAIAN	BOBOT NILAI
(1)	(2)	(3)	(4)	(5)	(6)	(7)
		3. RGB Dan HSV Perihal Himpunan				
3	Mampu mengaplikasikan algoritma garis dan poligon	Algoritma garis dan poligon 1. Algoritma Pembentukan Garis 2. Algoritma Garis Bresenham 3. Membuat Poligon	Ceramah, Latihan, Diskusi	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	5%
4	Mampu mengaplikasikan algoritma lingkaran dan ellips	Algoritma dan Lingkaran Ellips 1. Algoritma Pembuatan Lingkaran 2. Algoritma Elips	Ceramah, Latihan, Diskusi	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10%
5	Mampu menerapkan algoritma pembuatan kurva	Algoritma pembuatan kurva 1. Kurva 2. Algoritma Bezier	Ceramah, Latihan, Diskusi	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10%
6	Mampu mengaplikasikan algoritma pewarnaan bidang	Algoritma pewarnaan bidang	Ceramah, Latihan, diskusi, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10%

PERTEMUAN KE-	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN (MATERI AJAR)	METODE PEMBELAJARAN	PENGALAMAN BELAJAR MAHASISWA	KRITERIA PENILAIAN	BOBOT NILAI
(1)	(2)	(3)	(4)	(5)	(6)	(7)
		1. Pengertian Pewarnaan Bidang 2. Algoritma Pewarnaan Bidang				
7	Mampu membuat proyeksi geometri bidang	Proyeksi geometri bidang 1. Pengertian Proyeksi Geomentri 2. Traksonomi Proyeksi Geomentri 3. Proyeksi Paralel	Ceramah, Latihan, diskusi, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10%
UTS						
8	Mampu menerapkan Transformasi, translasi, dan rotasi	Transformasi, translasi, dan rotasi 1. Pengertian Tranformasi 2. Translasi 3. Rotasi	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10 %
9	Mampu menerapkan Dilatasi	Dilatasi 1. Pengertian Dilatasi 2. Shear	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10 %

PERTEMUAN KE-	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN (MATERI AJAR)	METODE PEMBELAJARAN	PENGALAMAN BELAJAR MAHASISWA	KRITERIA PENILAIAN	BOBOT NILAI
(1)	(2)	(3)	(4)	(5)	(6)	(7)
10	Mampu menerapkan viewing dan Klipping	Viewing dan Klipping 1. Viewing 2. Clipping	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10 %
11	Mampu menerapkan teknik antiliasing	Teknik Antialiasing	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	10 %
12	Mampu mahasiswa mampu menerapkan grafika 3 dimensi.	Grafika 3 dimensi. 1. Konsep Grafika 3D 2. Berbagai Proyeksi Dalam Grafika 3D	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	5 %
13	Mampu merancang pemrograman komputer grafik menggunakan API.	Pemrograman Komputer Grafik menggunkan API 1. OpenGL 2. Pengenalan Pengolahan Citra Digital	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	5 %

PERTEMUAN KE-	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN (MATERI AJAR)	METODE PEMBELAJARAN	PENGALAMAN BELAJAR MAHASISWA	KRITERIA PENILAIAN	BOBOT NILAI
(1)	(2)	(3)	(4)	(5)	(6)	(7)
14	Mampu membuat aplikasi grafika komputer.	Aplikasi grafika komputer.	Ceramah, Latihan, Tugas	Mahasiswa melakukan simulasi	Algoritma berjalan dengan baik	5 %
UAS						

Referensi

Donald D. Hearn, M. Pauline, Warren Carithers, 2011, Computer Graphics with Open GL (4th Edition), Prentice-Hall,

Vince, John A., 2010,Mathematics for Computer Graphics (Undergraduate Topics for Computer Science). 3rd. Springer.

Hong Zang dan Y. Daniel Liang, 2006, Computer Graphics using Java 2D and 3D, Prentice Hall.

Hill, F.S., 2001, Computer Graphics Using OpenGL, Prentice Hall, Upper Saddle River, NJ. ebook Irfan Maliki, S.T – Grafika Komputer ilmu Komputer
Dr. Setiawan Hadi - Modul Diktat Grafika Komputer

Pat Hanrahan, “Introductory Computer Graphics”, https://graphics.stanford.edu/courses/#cs148
-, “The Java Tutorials”, http://download.oracle.com/javase/tutorial/
-, ”Trail: 2D Graphics (The Java Tutorial)”, http://download.oracle.com/javase/tutorial/2d/index.html
-, “Java 3D API Tutorial”, http://java.sun.com/developer/onlineTraining/java3d/

Tangsel, 12 Maret 2021

Ketua Program Studi
S1 Teknik Informatika

Ketua Tim Penyusun
Mata Kuliah Komputer Grafik 1

Achmad Udin Zailani, S.Kom.,M.Kom
NIDN. 0429058303

Achmad Udin Zailani, S.Kom.,M.Kom
NIDN. 0429058303