

VOT 71843

**THE DEVELOPMENT OF REAL-TIME DAYLIGHT SKY COLOUR
SIMULATION FOR VIRTUAL ENVIRONMENT**

**(PEMBANGUNAN SIMULASI WARNA LANGIT SIANG SECARA
MASA NYATA UNTUK PERSEKITARAN MAYA)**

**MOHD SHAHRIZAL SUNAR
ABDULLAH BADE**

**RESEARCH VOTE NO:
71843**

**JABATAN GRAFIK KOMPUTER DAN MULTIMEDIA
FAKULTI SAINS KOMPUTER DAN SISTEM MAKLUMAT
UNIVERSITI TEKNOLOGI MALAYSIA**

2005

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : THE DEVELOPMENT OF REAL-TIME DAYLIGHT SKY
COLOUR SIMULATION FOR VIRTUAL ENVIRONMENT

Saya **MOHD SHAHRIZAL BIN SUNAR**
(HURUF BESAR)

mengaku membenarkan Laporan Akhir Penyelidikan ini disimpan di Perpustakaan Universiti
Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut:

1. Laporan Akhir Penyelidikan adalah hak milik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan Laporan Akhir Penyelidikan ini bagi kategori **TIDAK TERHAD**.
4. *Sila tandakan (✓)

SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

(TANDATANGAN KETUA PROJEK)

Nama & Cop Ketua Penyelidik
Tarikh: 15/11/2005

ABSTRAK

Grafik komputer kebiasaannya digunakan untuk menghasilkan simulasi dunia sebenar dengan penjana imej yang realistik, terutamanya untuk persekitaran luaran yang memerlukan langit sebagai latar belakang utama. Kajian penyelidikan ini bertujuan untuk menjana simulasi warna langit di waktu siang yang dapat memodelkan langit daripada sebarang lokasi di muka bumi. Kajian yang dilakukan hanya tertumpu kepada keadaan warna pada langit sahaja tanpa mengambil kira objek-objek lain yang terdapat pada langit seperti awan, pelangi, kesan atmosfera dan sebagainya. Kaedah dom langit atau hemisfera digunakan dalam pembangunan simulasi ini. Antara perkara-perkara yang penting dalam menentukan kedudukan matahari ialah waktu, tarikh dan lokasi pemerhati. Lokasi pemerhati ditentukan berdasarkan longitud dan latitud muka bumi. Model dom digunakan sebagai perwakilan kepada langit. Formula matematik asas untuk sfera digunakan untuk memodelkan struktur data yang mantap bagi model dom. Warna langit pula dikira dengan menggunakan model analitik bagi langit. Warna langit yang telah dikira akan dipetakan kepada verteks-verteks dalam dom. Kemudian warna di antara setiap verteks akan dikira secara interpolasi dan dilorekkan secara halus bagi menghasilkan anggaran warna bagi langit. Teknik ini telah dapat menaikkan lagi prestasi dan melajukan penghasilan simulasi. Justeru, ia adalah sangat sesuai bagi dilaksanakan terhadap aplikasi masa nyata dan juga dapat memberikan sumbangan terhadap model cuaca dalam permodelan persekitaran.

ABSTRACT

Computer graphics has frequently been used to produce a simulation of the real world by synthesising photo realistic image, especially for outdoors scenes, where displaying the sky as the background is indispensable. This research presents the sky dome or hemispheric approach in developing a daylight sky simulator, which endeavours to model the colour of the sky for any location of the planet. This research is only focus on the sky colour without clouds, rainbow or other atmosphere effect. A specific time of the day, the date and location of the viewer is needed to compute the position of the sun. The longitude and latitude is used to specify the viewer's position. A model of a dome is chosen as a representation of the sky. The basic mathematical formula for a sphere is used to model a robust data structure for the dome. The sky colour is computed using an analytic model of the sky. The calculated colour of the sky is mapped into the dome vertices; then the colour between each vertex is interpolated using smooth shading, which will give the approximate colour of the sky. The technique will speed up the rendering time. The result is suitable for rendering in a real-time application, which will contribute to the state-of-the-art for the climate models in environmental modelling.

KANDUNGAN

BAB	PERKARA	MUKA SURAT
	JUDUL	i
	PENGAKUAN	ii
	DEDIKASI	iii
	PENGHARGAAN	iv
	ABSTRAK	v
	ABSTRACT	vi
	KANDUNGAN	vii
	SENARAI RAJAH	xi
	SENARAI ISTILAH	xiii
	SENARAI JADUAL	xiv
	SENARAI LAMPIRAN	xv
BAB I	Pengenalan	
	1.1 Pendahuluan	1
	1.2 Penyataan Masalah	2
	1.3 Matlamat	4
	1.4 Objektif	4
	1.5 Skop	5
	1.6 Justifikasi Dan Kepentingan	5

1.7	Penutup	7
-----	---------	---

BAB II KAJIAN LITERATUR

2.1	Pendahuluan	8
2.2	Keadaan Langit	8
	2.2.1 Cahaya Matahari	10
	2.2.2 Cahaya Langit	11
2.3	Longitud, Latitud dan Sinaran Matahari	11
	2.3.1 Zon Masa	12
2.4	Serakan Cahaya	13
	2.4.1 Teori Serakan Cahaya	13
	2.5.1 Serakan Rayleigh	14
	2.5.2 Serakan Mie	15
2.5	Teori Warna	16
	2.5.1 Apa itu Warna	16
	2.5.2 Ruang Warna	17
	2.5.2.1Warna Kelabu	17
	2.5.2.2Warna RGB	18
	2.5.2.3Warna CMYK	19
	2.5.5.4Warna CIE	20
	2.5.2.5Warna CIE – XYZ	20
	2.5.2.6Warna CIE – Yxy	21
	2.5.3 Pertukaran dan Permodelan Nilai Warna	22
2.6	Hasil kajian lepas	24
2.7	Penutup	25

BAB III METODOLOGI

3.1	Pengenalan	26
3.2	Rangka Kerja Teori	27
3.3	Perlasanaan Kajian	28
3.3.1	Fasa I : Kajian terhadap Algoritma Warna Langit	28
3.3.2	Fasa II : Rekabentuk Teknik Simulasi Warna Langit	29
3.3.2.1	Permodelan langit	29
3.3.2.2	Algoritma Warna Langit	30
3.3.3	Fasa III : Pengkodan Aplikasi Simulasi Warna Langit	30
3.3.3.1	Aliran Aturcara	32
3.3.3.2	Rekabentuk Objek Kelas	33
3.3.3.3	Rekabentuk Antaramuka Pengguna	33
3.3.4	Fasa IV : Pengujian	34
3.3.5	Fasa V : Dokumentasi	35
3.3.5.1	Perancangan Masa	35
3.4	Analisa Keperluan Pembangunan	36
3.4.1	Keperluan Perkakasan secara minimum	36
3.4.2	Keperluan Perkakasan	37
3.5	Penutup	38

BAB IV IMPLIMENTASI

4.1	Pengenalan	39
4.2	Penghasilan Kedudukan Matahari	39
4.3	Penghasilam Model Langit Hemisfera	42
4.4	Penukaran Nilai Warna	53
4.4.1	Yxy kepada XYZ	53
4.4.2	XYZ kepada RGB	54

4.4.3	Pembangunan Perisian Penukaran Khas Warna	59
4.5	Pengiraan Warna Langit	60
4.5.1	Model Perez	60
4.6	Antaramuka Pengguna Bergrafik	65
4.7	Penutup	68

BAB V HASIL DAN PENGUJIAN

5.1	Pengenalan	69
5.2	Hasil Kajian	70
5.2.1	Perbandingan warna langit	70
5.3	Pengujian Pengiraan Kedudukan Matahari	80
5.4	Pengujian Penukaran Warna	81
5.5	Penutup	82

BAB VI PERBINCANGAN DAN KESIMPULAN

6.1	Pengenalan	83
6.2	Pencapaian kajian	83
6.2	Cadangan	86
6.3	Kesimpulan	88

BIBLIOGRAFI

LAMPIRAN

SENARAI RAJAH

NO. RAJAH	TAJUK	MUKA SURAT
2.0	Langit cerah tanpa awan	9
2.1	Langit yang diselubungi awan	9
2.2	Langit yang cerah tetapi mempunyai sedikit awan	10
2.3	Garisan longitud dan latitud bumi	12
2.4	Teori serakan cahaya	14
2.5	Teori serakan cahaya Rayleigh	14
2.6	Teori serakan cahaya Mie	15
2.7	Perbandingan teori Rayleigh dan Mie	16
2.8	Kategori warna kelabu	17
2.9	Kategori warna RGB	18
2.10	Kategori warna bagi HSV dan HLS	19
2.11	Kategori warna CIE – Yxy	20
3.0	Rangka kerja teori untuk projek Permodelan Warna Langit	27
3.1	Aliran aturcara projek	32
4.0	Algoritma menentukan kedudukan Matahari	42
4.1	Model Hemisfera	43
4.2	Gambaran sudut bagi zenith θ dan azimuth ϕ	45
4.3	Gambaran titik – titik yang di hasilkan	45
4.4	Keadaan dalam dan luar wireframe bagi hemisfera langit	48
4.5	Algoritma menghasilkan hemisfera	51
4.6	Algoritma penukaran warna CIE kepada Yxy kepada CIE-XYZ dan dari CIE XYZ kepada RGB	57
4.7	Penskalaan untuk nilai RGB	58

4.8	Screen captured bagi perisian penukaran ruang warna	59
4.9	Algoritma mengira sudut antara matahari dengan mana – mana titik di dalam hemisfera	61
4.10	Algoritma menentukan nilai untuk zenith Y, x dan y serta parameter untuk A,B,C,D dan E	64
4.11	Mod skrin penuh bersama dengan pemandangan Aerial	67
5.1	Eiffel Tower di Paris diambil pada 4 September 2001 jam 1940	72
5.2	Warna langit yang dijana untuk Eiffel Tower di Paris diambil pada 4 September 2001 jam 1940	72
5.3	Pantai Bridlington, England pada 4 September 2001 jam 1840	73
5.4	Warna langit yang dijana untuk Pantai Bridlington, England pada 4 September 2001 jam 1840	73
5.5	Gambar di kawasan Universiti Teknologi Malaysia pada 28 Julai 2002 jam 0845	74
5.6	Gambar yang diambil daripada aturcara yang dihasilkan pada 28 Julai jam 0845	74
5.7	Gambar di kawasan Universiti Teknologi Malaysia pada 28 Julai 2002 jam 0830	75
5.8	Gambar yang diambil daripada aturcara yang dihasilkan pada 28 Julai jam 0830	75
5.9	Gambar di sekitar kawasan Universiti Teknologi Malaysia pada 15 Ogos 2002 jam 0900	76
5.10	Gambar yang diambil daripada aturcara yang dihasilkan pada 15 Ogos 2002 jam 0900	76
5.11	Gambar di sekitar kawasan Universiti Teknologi Malaysia pada 15 Ogos 2002 jam 1600	77
5.12	Gambar yang diambil daripada aturcara yang dihasilkan pada 15 Ogos 2002 jam 1600	77
5.13	Gambar di sekitar kawasan Desa Skudai pada 17 Ogos jam 1500	78
5.14	Gambar yang diambil daripada aturcara yang dihasilkan pada 17 Julai jam 1500	78
5.15	Graf perbandingan di antara matahari terbit dan terbenam	79

SENARAI ISTILAH

Zenith	-	Sudut yang menghadap keatas
Azimuth	-	sudut yang mengelilingi permukaan dari horizon
Gradient	-	Perubahan warna
Solar declination	-	Perubahan kepada kedudukan matahari kebawah
Dome	-	Berbentuk Hemisfera
Logitude	-	garis bujur pada peta
Latitude	-	garis lintang pada peta
Turbidity	-	nilai kekeruhan

SENARAI JADUAL

JADUAL	PERKARA	MUKA SURAT
4.1	Jadual Koordinat Monitor	56
5.1	Jadual perbandingan antara matahari terbit dan terbenam	77

SENARAI LAMPIRAN

LAMPIRAN	PERKARA	MUKA SURAT
A	Manual Pengguna	

BAB 1

PENGENALAN

1.1 Pendahuluan

Kajian mengenai langit telah dibuat sekian lama, namun ianya tidak dapat diaplikasikan ke dalam bidang grafik komputer secara menyeluruh kerana kompleksiti komputasinya. Fenomena langit sungguh unik disebabkan keadaannya yang mempunyai kesan yang berbeza-beza seperti pelangi, kabus dan sebagainya.

Kebanyakan dari kita pasti akan mengatakan bahawa langit berwarna biru. Sebenarnya langit tidaklah selalu berwarna biru. Langit dipilih sebagai latar belakang apabila kita memodelkan sebuah gunung. Begitu juga di dalam simulasi kapal terbang di mana langit merupakan latar belakang yang utama. Warna bagi langit berubah dalam beberapa keadaan. Sebagai contoh pada waktu pagi, kita dapat melihat langit berwarna biru dan berwarna putih apabila semakin hampir kepada kaki langit. Apabila menjelang waktu matahari terbenam, warna bagi langit akan kelihatan kuning kemerah–merahan apabila berada di kaki langit.

Fenomena semulajadi ini menjadi perhatian untuk menghasilkan satu keadaan langit yang benar-benar realistik untuk dihasilkan di dalam komputer. Penghasilan warna langit yang realistik dapat memberi gambaran sebenar kepada sesuatu pemandangan di dalam imej komputer.

Untuk menghasilkan warna yang realistik ini, beberapa kajian dilakukan ke atas kesan-kesan yang menyebabkan perubahan warna langit dan bagaimana sebenarnya terhasil warna langit tersebut.

Penghasilan langit adalah perkara utama bagi sesuatu aplikasi simulasi yang melibatkan persekitaran luar sebagai latarbelakang utama bagi sesuatu kawasan. Pemvisualan langit sangat bergantung kepada kesan cahaya matahari. Jarak juga dapat memberikan kesan pemvisualan yang berbeza-beza. Matlamat utama dalam grafik komputer adalah menghasilkan kesan sepertimana yang terdapat dalam dunia sebenar.

Beberapa kajian dilakukan terhadap sifat-sifat cahaya, teori warna, kandungan atmosfera, matahari dan penghasilan algoritma-algoritma yang sesuai untuk paparan pada skrin. Kajian mengenai penghasilan warna langit telah giat dilakukan dan beberapa teknik yang ditemui akan dibincangkan.

1.2 Penyataan Masalah

Cabaran utama di dalam grafik komputer adalah menghasilkan imej grafik yang realistik. Untuk mendapatkan hasil yang realistik seharusnya kajian yang terperinci ke

atas penghasilan warna pada langit perlu dilakukan dengan mengambil kira pelbagai sudut.

Kajian untuk menghasilkan fenomena langit yang realistik untuk diaplikasikan ke dalam grafik komputer telah giat dijalankan. Langit merupakan objek yang sering dijadikan sebagai latarbelakang. Di dalam kebanyakan aplikasi 3 dimensi yang kita lihat sekarang ini seperti permainan komputer memaparkan objek langit sebagai objek utama sekiranya permainan itu melibatkan karakter yang berada di persekitaran luar. Di dalam aplikasi simulasi penerbangan juga menjadikan objek langit sebagai latarbelakang yang utama.

Kaedah paling mudah untuk mendapatkan warna langit sebagai latarbelakang dalam persekitaran maya ialah dengan menetapkan warna biru di bahagian atas dan warna putih atau kuning di bahagian bawah. Proses interpolasi antara kedua-dua warna ini akan menghasilkan warna yang seakan-akan warna langit.

Kaedah yang biasanya digunakan di dalam permainan komputer 3D adalah pengteksturan imej langit ke atas model langit. Teknik ini digunakan kerana tidak memerlukan pemprosesan yang tinggi. Namun kesan ke atas langit tidaklah begitu realistik. Di samping itu, keadaan langit yang dihasilkan adalah statik bagi semua kawasan di muka bumi. Situasi ini akan menggambarkan keadaan langit yang konsisten tanpa ada sebarang perubahan pada masa dan tempat.

Terdapat kajian terdahulu terhadap warna langit yang dilakukan oleh Blinn(1982), Max(1986), Klassen(1987), Kaneda et al. (1991) dan Nishita et al. (1996). Hasil kajian kebanyakannya menghasilkan warna langit yang tepat tetapi mempunyai kekangan dalam komputasi. Ia tidak dapat dilaksanakan secara masa nyata.

Terdapat beberapa faktor yang mempengaruhi warna ke atas langit. Di antaranya ialah kedudukan matahari, kedudukan pemerhati dan keadaan atmosfera. Isu utama

yang ingin diselesaikan di dalam kajian ini ialah bagaimana untuk menghasilkan paparan warna langit secara masa nyata apabila diberikan kedudukan pemerhati berserta tarikh dan masa tanpa mengabaikan kualiti visual terhadap warna langit tersebut.

1.3 Matlamat

Matlamat projek ini adalah untuk menghasilkan simulasi warna langit waktu siang secara masa nyata untuk persekitaran maya.

1.4 Objektif

Untuk mencapai matlamat seperti yang telah dinyatakan beberapa objektif telah dihasilkan seperti di bawah:

1. Mendapatkan kedudukan matahari berdasarkan kepada kedudukan (longitud dan latitud), tarikh dan waktu pemerhati.
2. Menghasilkan model langit menggunakan hemisfera.
3. Menghasilkan simulasi perubahan warna langit berdasarkan perubahan kedudukan matahari dalam persekitaran maya

1.5 Skop

Skop bagi projek ini meliputi perkara-perkara seperti di bawah:

1. Tumpuan projek ini ialah untuk menghasilkan simulasi warna langit pada waktu siang.
2. Menganggap bahawa langit sentiasa dalam keadaan cerah dan tidak berawan.
3. Model langit yang dihasilkan tiada kesan-kesan atmosfera seperti pelangi, gerhana, jerebu dan sebagainya.
4. Tidak mengambil kira kesan-kesan seperti hujan, ribut dan puting beliung.
5. Pandangan pengguna boleh dilihat dari dalam dan luar atmosfera.

1.6 Justifikasi dan kepentingan

Dalam kajian ini, penghasilan persekitaran maya yang menyerupai keadaan yang sebenar dapat dipamerkan untuk penggunaan pelbagai aplikasi grafik. Sebagai contoh, aplikasi simulasi penerbangan dapat menggunakan hasil kajian ini untuk memaparkan imej langit yang realistik.

Aplikasi ini dapat membantu para penyelidik untuk mengetahui keadaan warna langit pada kawasan-kawasan tertentu. Dalam pada itu, ianya dapat digunakan dalam aplikasi grafik yang memerlukan langit sebagai imej untuk latarbelakang.

Untuk menghasilkan satu permainan komputer yang realistik, teknik ini dapat diimplimentasikan. Ianya dapat memberikan satu kesan realistik kepada persekitaran di dalam permainan komputer tersebut. Dalam pada itu, ianya juga boleh diimplimentasikan dalam permainan simulasi penerbangan.

Justeru, hasil kajian ini boleh digunakan oleh pelbagai pihak dalam pelbagai bidang. Antaranya adalah seperti pembangun simulator penerbangan, penghasilan visualisasi ramalan cuaca, pembangun permainan komputer dan juga perekabentuk senibina bangunan. Selain itu, hasil kajian ini juga dapat dimanfaatkan oleh penyelidik dalam bidang grafik komputer dan persekitaran maya itu sendiri

1.7 Penutup

Sebagai pengenalan untuk projek ini, satu penerangan tentang tajuk projek telah diberikan. Seterusnya, pernyataan masalah serta matlamat dan objektif projek ini telah dijelaskan. Skop dan limitasi kajian ini juga telah dibincangkan. Aplikasi dan kepentingan kajian ini telah dihuraikan dalam bahagian justifikasi dan kepentingan. Dalam bab yang seterusnya, kajian literatur yang berkaitan dengan kajian ini akan dibincangkan.

BAB 2

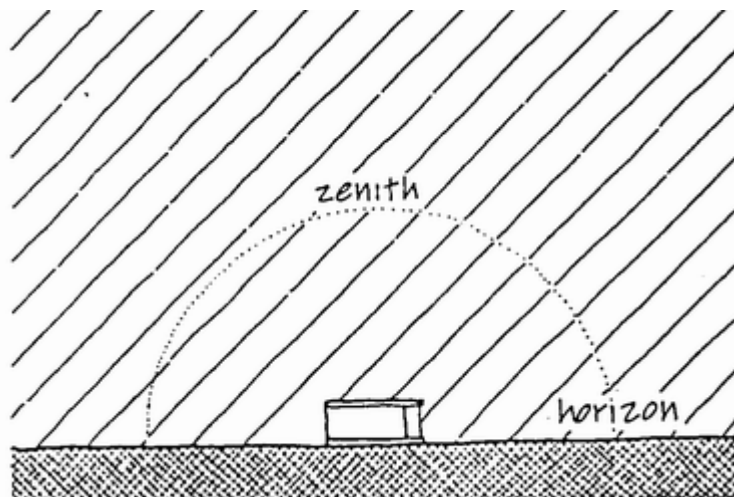
KAJIAN LITERATUR

2.1 Pendahuluan

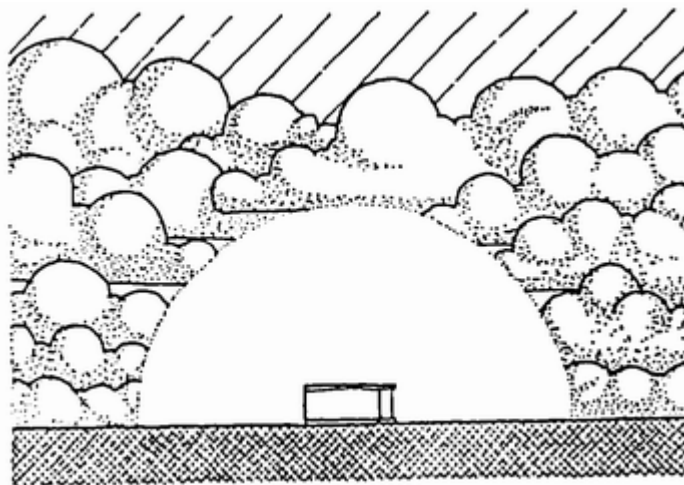
Di dalam bab ini, beberapa teori mengenai keadaan langit dimuatkan untuk mengenalpasti keadaan warna bagi langit yang dihasilkan. Fakta mengenai teori serakan dan kedudukan cahaya matahari yang menyebabkan berlakunya perubahan warna pada langit. Teori warna juga diterangkan bagi tujuan pemilihan warna yang bersesuaian untuk pembangunan projek ini. Pemilihan Java3D sebagai satu pustaka grafik yang ingin diimplimentasikan dalam projek ini juga akan diterangkan.

2.2 Keadaan Langit

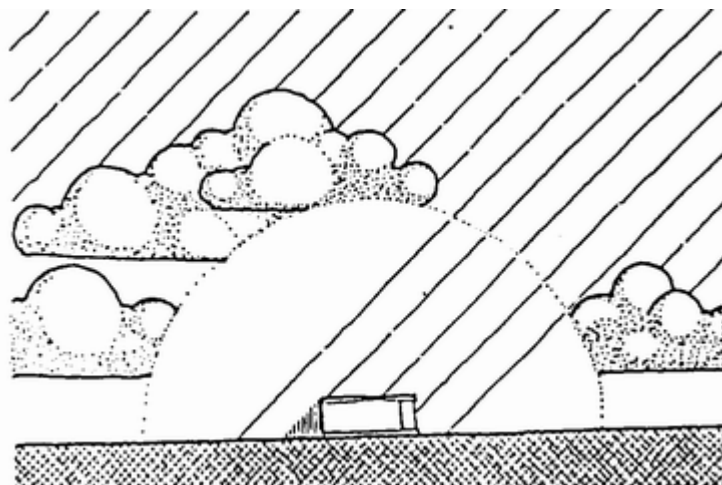
Langit pada waktu siang terbahagi kepada beberapa keadaan. Warna pada langit berbeza pada keadaan yang berbeza. Diantara keadaan-keadaan langit ialah langit yang cerah tanpa berawan, langit yang diselubungi penuh dengan awan dan langit yang cerah tetapi mempunyai sedikit awan.



Rajah 2.0 : Langit yang cerah tanpa berawan



Rajah 2.1 : Langit yang diselubungi penuh dengan awan



Rajah 2.2 : Langit yang cerah tetapi mempunyai sedikit awan

2.2.1 Cahaya Matahari

Cahaya matahari merupakan sumber cahaya yang paling utama kepada bumi. Ianya memancar secara selari kearah permukaan bumi. Cahaya matahari diserakkan oleh himpunan gas-gas dan partikel-partikel yang terdapat pada atmosfera bumi sebelum sampai kepada mata manusia. Cahaya yang mempunyai frekuensi yang tinggi dan panjang gelombang yang pendek akan diserakkan lebih daripada cahaya yang mempunyai frekuensi yang rendah dan panjang gelombang yang tinggi. Cahaya biru lebih banyak diserakkan kerana mempunyai panjang gelombang yang pendek. Oleh yang demikian, langit kelihatan berwarna biru.

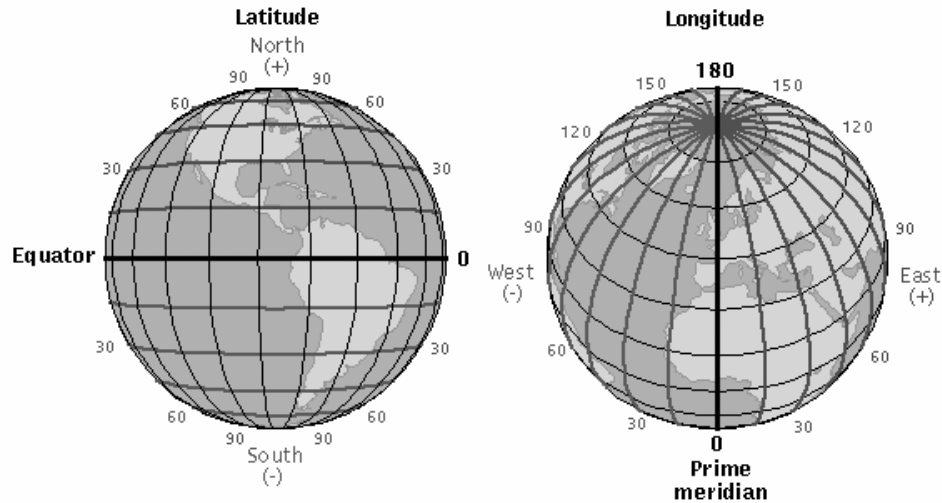
Keamatan cahaya matahari bergantung kepada tarikh, masa dan lokasi. Sebagai contoh, keamatan cahaya di kawasan tropika adalah tinggi pada waktu tengahari apabila matahari tegak di atas kepala, manakala keamatan cahaya adalah rendah pada musim sejuk di kawasan artik. Perubahan kedudukan keamatan cahaya matahari akan mempengaruhi warna pada langit mengikut teori Mie dan Rayleigh.

2.2.2 Cahaya Langit

Cahaya langit adalah cahaya yang disebabkan oleh kesan serakan cahaya matahari. Kesan daripada serakan itu tadi, menghasilkan cahaya yang menuju kepada pelbagai arah. Warna pada langit berubah bergantung kepada kedudukan matahari, kedudukan seseorang yang memerhati langit dan keadaan atmosfera. Cahaya langit ini memberikan keadaan warna yang berlainan pada langit disebabkan oleh serakan molekul–molekul udara.

2.3 Longitud, Latitud dan Sinaran Matahari

Untuk mengetahui kedudukan sesuatu tempat di muka bumi, satu sistem koordinat bumi telah diwujudkan. Koordinat bumi ini berdasarkan kepada garisan longitud yang menghala dari utara ke selatan, manakala garisan latitud yang melalui dari timur ke barat. Garisan–garisan maya yang diwujudkan ini dapat memberikan satu lokasi yang tepat dimuka bumi. Garis 0 untuk longitud terletak pada meridian Greenwich. Sebagai contoh, garisan latitud dan longitud untuk Johor Bahru adalah $1^{\circ} 38''\text{N}$, $103^{\circ} 40' \text{ E}$.



Rajah 2.3 : Garisan longitud dan latitud bumi

2.3.1 Zon Masa

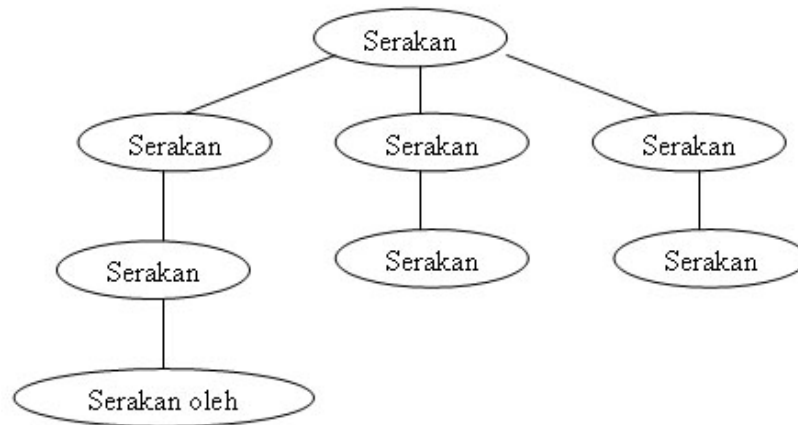
Bumi telah dibahagikan kepada 24 kawasan di mana setiap kawasan ditentukan oleh setiap zon masa untuk menentukan waktu yang piawai. Penggunaan zon masa ini dapat memberikan satu waktu yang piawai kerana bumi berbentuk bulat dan setiap waktu matahari terbit dan terbenam adalah berbeza. Bumi yang mempunyai 360 darjah dibahagikan kepada 24 kawasan di mana setiap kawasan mempunyai kelebaran 15 darjah.

2.4 Serakan Cahaya

Apabila cahaya mengenai kepada sesuatu bahan ianya akan bertindakbalas dengan atom-atom pada bahan tersebut dan kesan yang berlaku adalah bergantung kepada panjang gelombang cahaya dan keadaan atom pada bahan tersebut. Serakan cahaya berlaku apabila atom-atom pada bahan-bahan lutsinar tidak berada pada keadaan sekata pada jarak yang lebih besar daripada panjang gelombang cahaya tetapi menjadi sekumpulan molekul-molekul atau partikel-partikel. Langit kelihatan cerah kerana molekul-molekul dan partikel-partikel di udara telah menyerakkan cahaya matahari. Oleh yang demikian, warna yang sampai kepada mata manusia telah diserakkan oleh molekul-molekul dan partikel-partikel tadi. Beberapa teori mengenai kesan penyerakan cahaya yang disebabkan oleh molekul udara telah dilakukan oleh Rayleigh dan Mie. Kesan serakan cahaya matahari ini menghasilkan cahaya langit.

2.4.1 Teori Serakan Cahaya

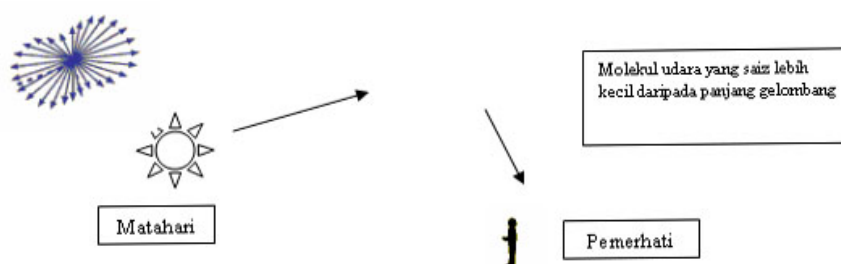
Banyak teori mengenai kesan serakan cahaya telah dikaji. Di antara teori serakan cahaya adalah teori Rayleigh, Mie, Brillouin, Bragg, Raman dan Compton. Serakan Rayleigh dan Mie adalah lebih fokus kepada kesan serakan cahaya ke atas molekul-molekul udara.



Rajah 2.4 : Teori serakan cahaya

2.4.2 Serakan Rayleigh

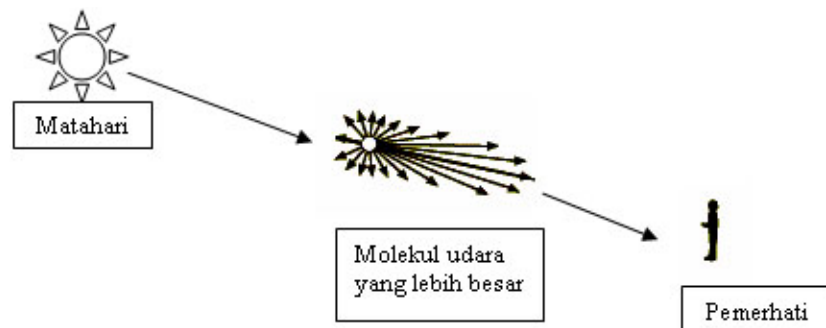
Serakan ini merujuk kepada kesan serakan cahaya matahari ke atas molekul pada udara. Rayleigh merupakan seorang ahli fizik British (1842 – 1919) mengira cahaya yang diserakkan oleh molekul udara yang mempunyai saiz yang lebih kecil daripada panjang gelombang. Mengikut teori Rayleigh, cahaya biru lebih banyak diserakkan kerana cahaya biru mempunyai panjang gelombang yang pendek di dalam spektrum warna. Oleh yang demikian, langit kelihatan berwarna biru.



Rajah 2.5 : Teori serakan cahaya Rayleigh

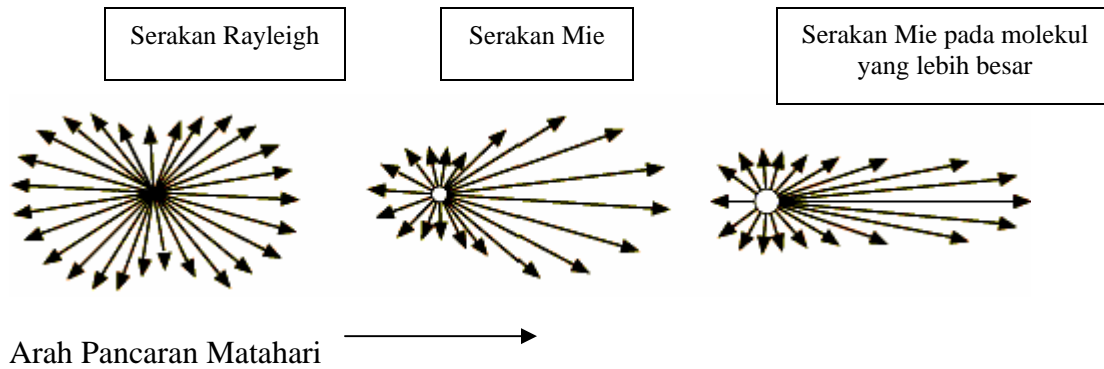
2.4.3 Serakan Mei

Sekiranya serakan Rayleigh hanya merujuk kepada kesan serakan cahaya ke atas molekul udara yang lebih kecil daripada panjang gelombang, serakan Mie pula menghuraikan mengenai kesan serakan cahaya ke atas molekul yang lebih besar. Teori Mie tidak mengambil kira kesan serakan oleh panjang gelombang. Ini kerana serakan Mie merujuk kepada kesan penghasilan cahaya berwarna putih berhampiran dengan matahari.



Rajah 2.6 : Teori serakan cahaya Mie

Perbandingan diantara serakan Mie dan Rayleigh akan menghasilkan keadaan warna pada langit sekiranya kita melihat kepada keadaan yang berlainan. Keadaan ini menghasilkan kesan warna yang berlainan pada langit apabila kita melihat berhampiran dengan matahari.



Rajah 2.7 : Perbandingan teori Rayleigh dan Mie

2.5 Teori Warna

2.5.1 Apa itu warna?

Warna adalah cahaya yang mempunyai panjang gelombang yang boleh dilihat oleh mata manusia. Panjang gelombang cahaya yang boleh dilihat oleh mata manusia adalah di antara 390 nanometer untuk cahaya berwarna ungu hingga 720 nanometer untuk cahaya berwarna merah. Jarak diantara kedua-dua panjang gelombang ini mempunyai pelbagai jenis warna yang dinamakan spektrum warna. Ruang warna adalah model menggambarkan warna dalam pelbagai kategori kecerahan sesuatu warna. Ianya menggambarkan bagaimana maklumat sesuatu warna itu dipersembahkan. Warna dapat dipersembahkan dalam bentuk 2, 3 dan 4 dimensi. Secara visualnya kebiasaanya warna akan dipersembahkan dalam bentuk kiub, kon dan polyhedra. Terdapat beberapa kategori warna yang digunakan dalam perkakasan yang berlainan. Sebagai contoh kategori warna RGB (merah, hijau dan biru) digunakan dalam monitor sementara

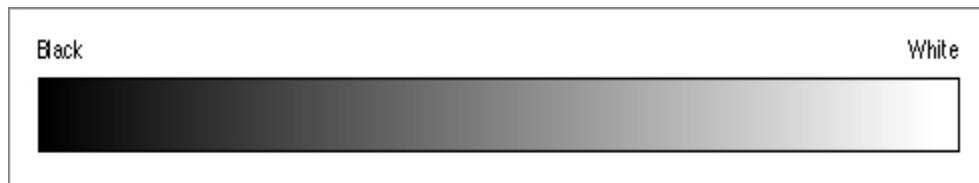
kategori warna yang digunakan dalam pencetak adalah kategori warna CMYK (cyan, magenta, yellow and black). Secara umumnya kategori warna yang dapat dipersembahkan dalam beberapa kategori adalah:

- kategori warna kelabu, digunakan oleh monitor dan pencetak
- kategori warna RGB, digunakan oleh pengimbas dan monitor
- kategori warna CMYK, digunakan untuk pencetak
- kategori warna tanpa perkakasan, diwakili oleh warna yang sebenar yang dilihat oleh mata manusia.

2.5.2 Ruang warna

2.5.2.1 Warna kelabu

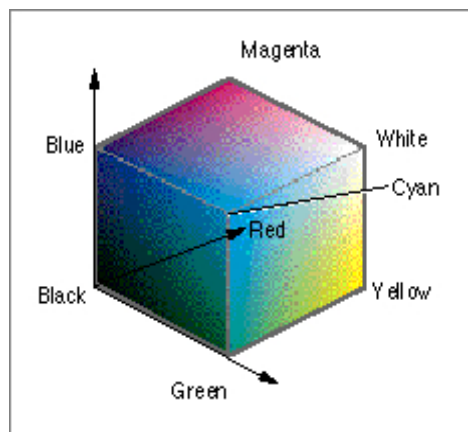
Kategori warna kelabu terdiri daripada perubahan warna hitam kepada warna putih.



Rajah 2.8 : Kategori warna kelabu

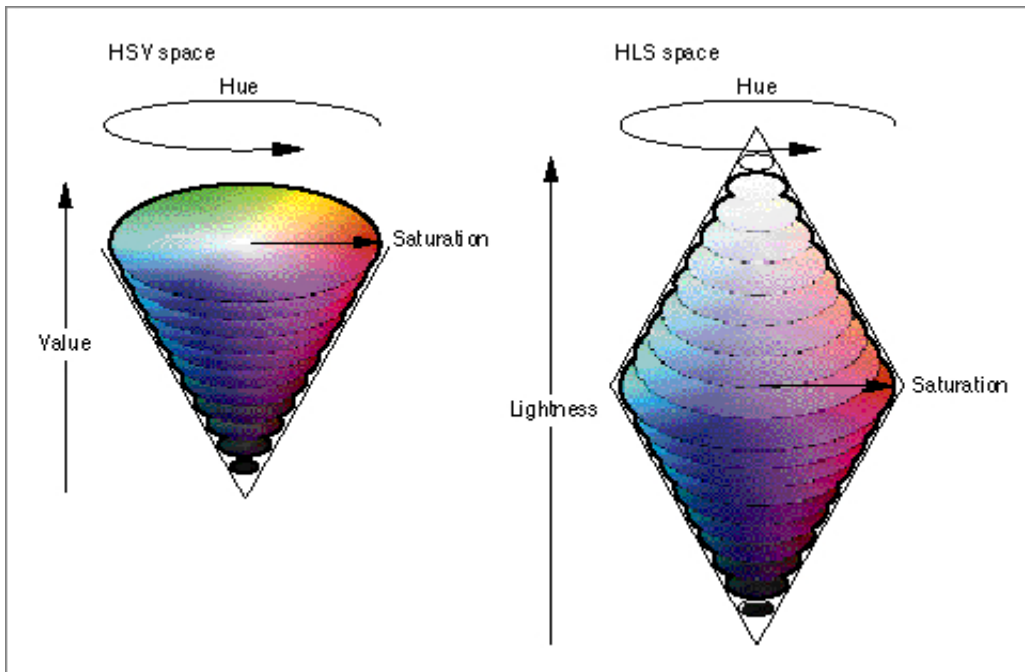
2.5.2.2 Warna RGB

Kategori warna RGB adalah warna yang diwakili dalam bentuk 3 dimensi di mana komponen utamanya ialah merah, hijau dan biru. Warna RGB adalah warna yang biasa digunakan di dalam komputer grafik. Komponen warna RGB bergantung kepada monitor di mana sekiranya monitor yang berbeza akan menghasilkan komponen warna RGB yang berbeza.



Rajah 2.9 : Kategori warna RGB

Transformasi daripada warna RGB dengan menghasilkan warna yang lebih semulajadi kepada seseorang pelukis adalah termasuk di dalam kategori warna HSV (Hue, Saturation dan Value) dan HLS (Hue, Lightness dan Saturation). Kedua – dua kategori warna ini boleh digambarkan dengan kon dan kon berkembar.



Rajah 2.10 : Kategori warna bagi HSV dan HLS

2.5.2.3 Warna CMYK

Kategori warna CMYK adalah warna yang biasa digunakan di dalam pencetak. CMYK merujuk kepada warna cyan, magenta, kuning dan hitam. Pada keadaan sebenar proses untuk menukarkan warna yang terdapat pada monitor dengan menggunakan kategori warna RGB tidak sama dengan warna yang digunakan pada pencetak yang menggunakan kategori warna CMYK. Apabila sesuatu gambar dicetak, kita dapat melihat perbezaan gambar pada monitor dan gambar yang dicetak di atas kertas. Keadaan ini disebabkan oleh warna CMYK bergantung kepada jenis pencetak, dakwat dan kertas yang digunakan.

2.5.2.4 Warna CIE

Kebanyakan warna yang dihasilkan adalah merujuk kepada peralatan yang tertentu. Sebagai contoh warna RGB hanya digunakan oleh pengimbas dan monitor manakala warna CMYK digunakan oleh pencetak untuk menghasilkan gambaran warna yang sebenar. Peralatan yang berlainan akan menghasilkan keamatan warna yang berlainan.

Untuk menghasilkan kategori warna yang tidak bergantung kepada mana-mana peralatan dan seperti yang dilihat oleh manusia, satu kategori warna telah dihasilkan. Hasil daripada kerja-kerja yang telah dilakukan pada tahun 1931 oleh Commission Internationale d'Eclairage (CIE) telah menghasilkan satu kategori warna yang dikenali sebagai kategori warna CIE.

Kategori warna CIE adalah spesifik kepada warna yang diterima oleh penglihatan manusia. Satu algoritma telah dihasilkan untuk menggambarkan secara 3 dimensi komponen warna X, Y dan Z di mana apabila ianya digabungkan akan menghasilkan semua warna yang boleh dilihat oleh mata manusia. Kategori warna ini dapat mengekalkan warna yang konsisten menerusi pelbagai peralatan berbanding dengan warna RGB dan CMYK.

2.5.2.5 Warna CIE-XYZ

Warna CIE-XYZ, adalah penggabungan warna daripada komponen warna X, Y dan Z yang mana ianya akan menghasilkan kombinasi warna yang diterima oleh retina mata manusia. Daripada komponen asal ini, warna ini dipecahkan lagi kepada komponen warna CIE-Yxy.

2.5.2.6 Warna CIE-Yxy

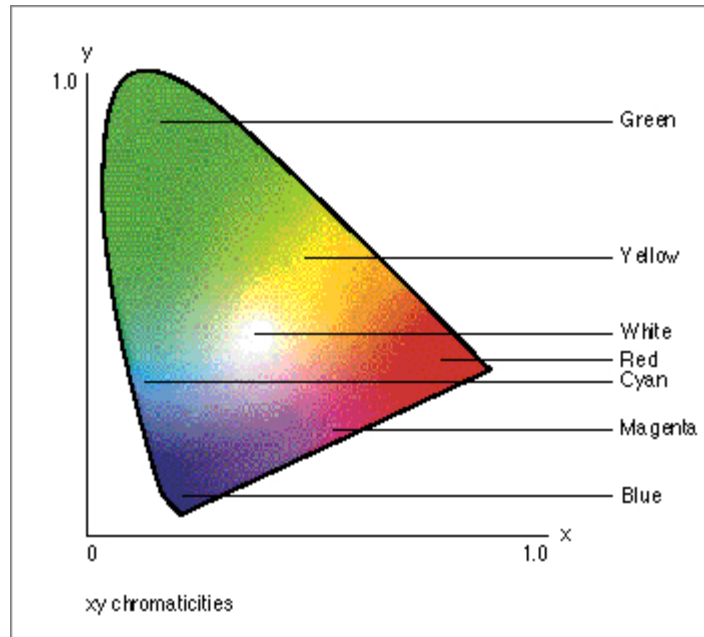
Warna CIE-Yxy adalah warna yang diambil daripada komponen warna XYZ tetapi ianya hanya menggambarkan nilai XYZ dalam bentuk koordinat x dan y. Nilai Z telah ditakrifkan dalam bentuk x dan y.

$$Y = Y$$

$$x = X / (X+Y+Z)$$

$$y = Y / (X+Y+Z) \dots\dots\dots(2.1)$$

Rajah di bawah menunjukkan warna Yxy



Rajah 2.11 : Kategori warna CIE – Yxy

2.5.3 Pertukaran dan pepadanan nilai warna

Penukaran warna adalah proses menukarkan warna dari satu ruang ke warna yang lain. Sementara penyepadanan warna melibatkan proses penyesuaian warna yang ditukar. Proses penyepadanan disesuaikan dengan warna untuk mencapai kesamaan secara maksimum dari ruang warna gamut kepada warna yang lain. Penyepadanan warna memerlukan penukaran warna, jika tidak pertukaran warna mungkin tidak berjaya.

Seperti yang diterangkan sebelum ini bahawa peralatan yang berlainan akan menghasilkan keamatan warna yang berbeza. Dari sekarang, setiap proses memerlukan gamut yang berlainan. Setiap monitor menggunakan warna RGB, namun setiapnya menggunakan RGB gamut yang berlainan. Pencetak juga salah satu peralatan yang bekerja dalam ruang CMYK dan berubah secara drastik dalam gamutnya. Ia berubah

terutama apabila menggunakan pencetak yang berlainan. Bagi memperbaiki imej yang dipaparkan pada pencetak yang berlainan, warna dipindahkan dengan cermat.

Rajah 2.6 menunjukkan dua peralatan menggunakan warna gamut yang ditukar kepada warna Yxy. Kedua-dua peralatan menghasilkan kurang daripada jumlah yang sebenar julat warna dan gamut pencetak adalah terhad kepada julat yang kecil yang tidak tetap berbanding gamut RGB.

Oleh itu, masalah ketidaksepadanan yang berlaku memaparkan imej yang sama untuk kedua-dua peralatan. Cara mengatasinya adalah dengan sedapkan warna dengan imej warna menggunakan data untuk kedua-dua peralatan atau modul pengurusan warna.

Dalam situasi yang berbeza, untuk memindahkan antara warna CIE kepada warna RGB, transformasi warna diperlukan. Namun, transformasi matrik bergantung kepada perbuatan fosfor tertentu dalam monitor yang tertentu. Oleh itu, matrik transformasi dibina untuk mencapai ketepatan pertukaran warna. Kadang kala nilai CIE XYZ akan memberi nilai negatif apabila berlaku perpindahan kepada warna RGB, tetapi sebaliknya bagi nilai RGB tidak akan menjadi nilai negatif. Oleh sebab itu, proses penyepadanan warna diperlukan. Dari sekarang, sesetengah warna yang digambarkan ke dalam warna CIE tidak boleh digambarkan ke dalam warna RGB.

2.6 Hasil kajian lepas

Kajian terhadap warna langit telah banyak dilakukan oleh penyelidik grafik. Kebanyakan mereka menghasilkan simulasi warna pada langit dengan mengambil kira kesan serakan cahaya ke atas molekul pada atmosfera. Beberapa teknik yang telah digunakan sebelum ini dikenalpasti untuk menghasilkan permodelan warna yang dilihat pada langit.

Klassen (1987) telah menghasilkan warna langit dengan mengambil kira kesan serakan cahaya oleh partikel di atmosfera. Bagaimanapun cara ini menghadapi masalah kerana atmosfera menyerupai beberapa lapisan selari dan mempunyai keseragaman dari segi kandungannya. Cara ini berbeza daripada keadaan fenomena yang sebenar dan ianya tidak menghasilkan kesan yang realistik.

Kaneda (1991) telah memperbaiki teknik yang digunakan oleh Klassen dengan menyerupai keadaan sebenar atmosfera. Kaneda telah mengandaikan bahawa kandungan atmosfera yang berbentuk sfera berkadaran dengan kandungannya berdasarkan kepada altitud.

Teknik yang dihasilkan oleh Kaneda telah diperbaiki oleh Nishita (1996) dengan menggunakan kaedah *multiple scattering*. Teknik-teknik yang dihasilkan di atas memerlukan kuasa pemrosesan yang tinggi dan memerlukan masa untuk penjanaan proses.

Dobashi (1997) ingin menemukan satu teknik menghasilkan warna langit dengan menggunakan fungsi asas. Walau bagaimanapun, keadaan langit yang dihasilkan dengan menggunakan fungsi ini tidaklah begitu realistik.

Preetham (1999) di dalam kajiannya telah mengemukakan kaedah model analitik. Imej yang dihasilkan sungguh mengagumkan. Dalam teknik ini, Preetham telah menggunakan kategori warna CIE-Yxy yang tidak mempengaruhi apa-apa jenis perkakasan. Dalam kajiannya, ia dapat menghasilkan satu keadaan warna langit yang mempunyai keadaan atmosfera yang berbeza-beza.

2.7 Penutup

Dalam bab ini, segala aspek-aspek mengenai kajian telah dilakukan untuk mengenalpasti keadaan-keadaan yang menyebabkan terjadinya warna pada langit. Pelbagai faktor-faktor telah dikenalpasti iaitu dari segi sifat serakan cahaya, keadaan atmosfera, teori serakan cahaya, teori warna dan teknik-teknik yang digunakan sebelum ini untuk menghasilkan warna pada langit. Kajian-kajian ini diperlukan untuk memberi gambaran yang jelas bagaimana terjadinya warna pada langit. Penggunaan JAVA 3D sebagai antaramuka pengaturcaraan aplikasi yang ingin dihasilkan dalam projek ini juga diterangkan.

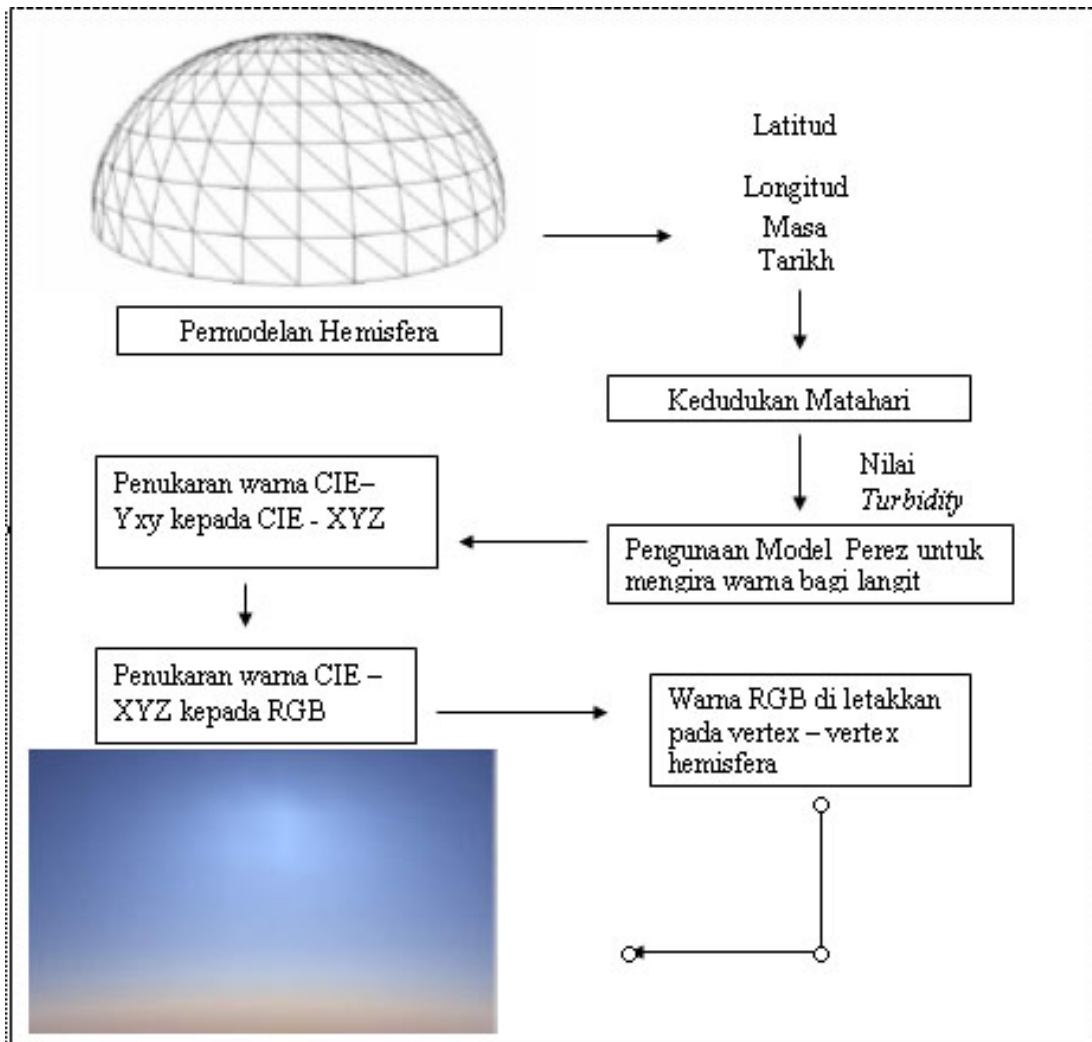
BAB 3

METODOLOGI

3.1 Pendahuluan

Di dalam bab ini, beberapa fasa-fasa yang diperlukan untuk menerangkan bagaimana projek ini dilaksanakan. Pada setiap fasa dihuraikan mengenai aktiviti yang akan dilakukan. Rangka kerja teori ditunjukkan untuk memberikan gambaran secara visual dalam menghasilkan permodelan warna langit.

3.2 Rangka Kerja Teori



Rajah 1.0 : Rangka kerja teori untuk projek permodelan warna langit

Sebuah model hemisfera dihasilkan bagi tujuan penghasilan objek untuk langit. Model ini sesuai digunakan kerana ianya dapat menggambarkan keadaan langit yang mengelilingi pemerhati. Bentuk hemisfera sering kali digunakan dalam pemodelan langit kerana dapat menggambarkan situasi sebenar dan pandangan bagi pemerhati akan diletakkan di dalam model hemisfera tersebut.

Lokasi bagi pemerhati yang berdasarkan kepada longitud dan latitud, masa dan juga tarikh diperlukan sebagai input. Input-input bagi data tersebut akan digunakan bagi tujuan pengiraan kedudukan matahari. Kedudukan matahari berbeza pada setiap bulan dan setiap kedudukan. Untuk mendapatkan data-data input tadi, satu formula yang berdasarkan kepada kedudukan matahari digunakan.

Model Perez dipilih untuk proses pengiraan bagi warna langit. Pengiraan akan dihasilkan untuk setiap titik pada model hemisfera. Pengiraan akan menggunakan kategori warna CIE $-Y_{xy}$ dan kemudiannya akan ditukarkan kepada kategori warna CIE-XYZ.

Untuk paparan pada skrin monitor, kategori warna CIE-XYZ akan ditukarkan kepada kategori warna RGB. Warna bagi langit akan bergantung kepada kesan kekeruhan pada langit. Nilai ini diwakili dengan satu parameter yang dikenali sebagai *Turbidity*. Parameter ini akan mempengaruhi penghasilan warna langit. Rajah 3.0 menunjukkan rangka kerja teori bagi menghasilkan projek permodelan warna langit ini.

3.3 Perlaksanaan Kajian

3.3.1 Fasa I : Kajian Terhadap Algoritma Warna Langit

Kajian mengenai keadaan langit dilakukan untuk mengetahui keadaan langit yang sebenar. Pada fasa ini, keseluruhan aspek dikenalpasti terlebih dahulu untuk

memenuhi skop dan matlamat projek ini. Keadaan langit perlulah pada waktu siang dan tidak berawan. Pengetahuan mengenai longitud, latitud dan zon masa perlu untuk menentukan keadaan langit pada kawasan berkenaan. Pada fasa ini juga diterangkan mengenai teori warna. Beberapa algoritma warna langit yang telah dihasilkan sebelum ini dikaji dan pemilihan salah satu daripada algoritma yang bersesuaian dengan matlamat projek ini.

3.3.2 Fasa II : Rekabentuk Teknik Simulasi Warna Langit

Pada fasa ini, rekabentuk ke atas permodelan langit, penggunaan algoritma langit dan pemilihan pustaka grafik yang bersesuaian untuk memenuhi keperluan projek dirangka.

3.3.2.1 Permodelan Langit

Untuk membentuk langit, memerlukan sebuah model yang sesuai. Penggunaan model hemisfera atau separuh sfera sesuai digunakan. Hemisfera merupakan objek yang tidak menampakkan sebarang sudut. Disamping itu, ianya sesuai digunakan untuk memplotkan setiap titik warna langit ke atas hemisfera berkenaan berpandukan kepada algoritma warna langit yang akan digunakan.

3.3.2.2 Algoritma warna langit

Preetham (1999) telah memperkenalkan sebuah kaedah yang dapat menghasilkan warna langit yang realistik. Ianya berdasarkan kepada model Perez yang mengira *luminance* bagi langit dengan pelbagai keadaan. Teknik yang digunakan dapat mengurangkan masa pengiraan dan mempercepatkan penghasilan warna. Warna yang dihasilkan dapat menghampiri kepada keadaan sebenar. Walaupun ianya tidak dapat benar-benar menghampiri keadaan sebenar kerana keadaan langit sebenar tidak sama pada setiap masa. Kandungan aerosol pada langit yang tinggi dapat mempengaruhi warna langit sebenar. Warna terdiri daripada beberapa kategori, di dalam kaedah yang diperkenalkan ini, kategori warna CIE–Yxy digunakan kerana ianya tidak dipengaruhi oleh faktor–faktor lain. Untuk tujuan paparan di dalam komputer, maka proses penukaran warna CIE–Yxy kepada warna RGB perlu dilakukan.

3.3.3 Fasa III : Pengkodan Aplikasi Simulasi Warna Langit

Di antara matlamat projek ini adalah untuk menghasilkan simulasi warna langit mengikut kawasan, tarikh dan masa.

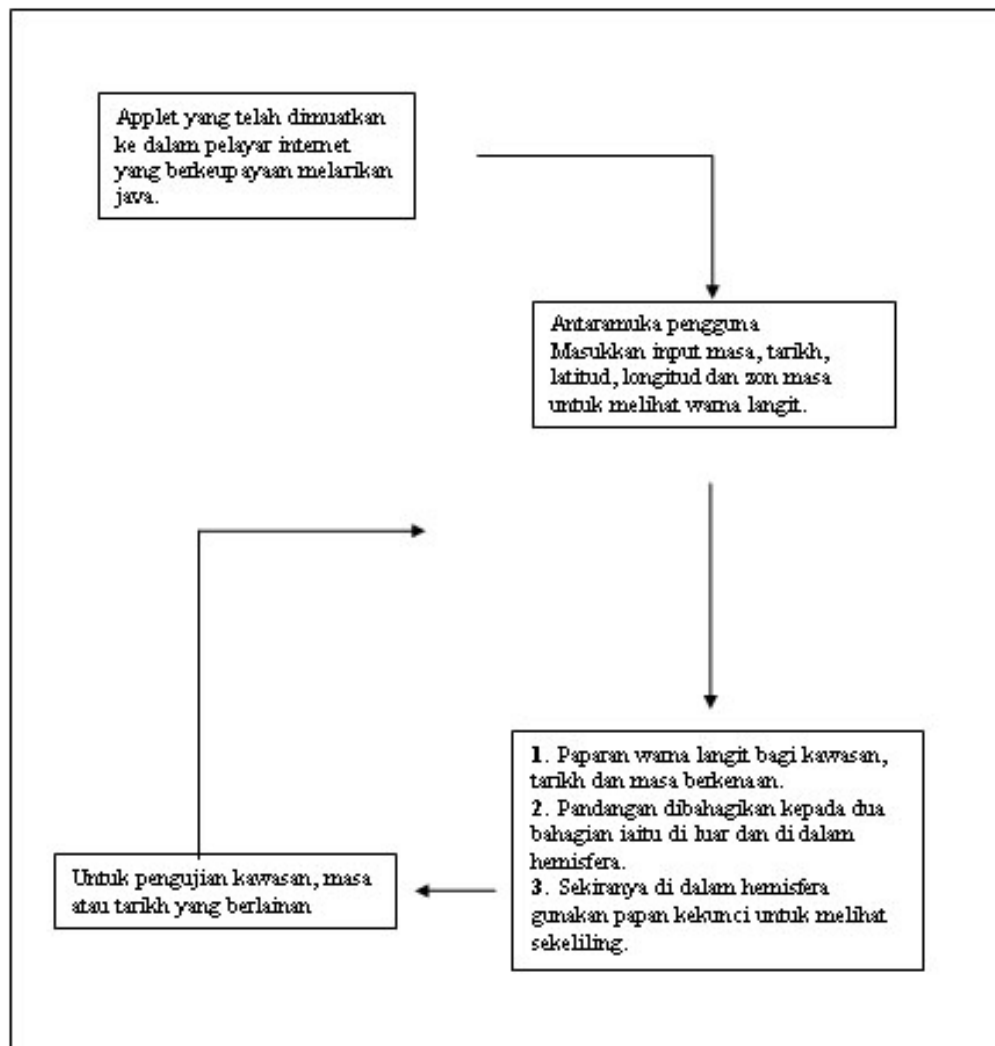
Untuk membolehkan setiap pengguna melihat keadaan langit bagi kawasan tertentu, satu antaramuka pengguna dihasilkan. Pengguna perlu mengetahui longitud, latitud dan zon masa bagi kawasan berkenaan. Pengguna juga dikehendaki meletakkan

input masa dan tarikh. Perubahan bagi langit dapat diperhatikan apabila tarikh yang berlainan pada kawasan dan waktu yang sama. Penghasilan antaramuka ini adalah sebahagian pengujian yang dilakukan untuk mendapatkan warna langit bagi kawasan, masa dan tarikh yang berlainan.

Hasil akhir adalah dalam bentuk applet yang akan dimuatkan ke dalam html di mana proses pengujian boleh dilakukan hanya dengan menggunakan pelayar yang mempunyai java. Carta alir di bawah menunjukkan bagaimana aliran aturcara.

Selain daripada menggunakan carta alir sebagai rujukan, rajah peralihan juga merupakan cara paling baik untuk menggambarkan rekabentuk program. Pada permulaan, data-data input seperti longitud, latitud, masa dan tarikh dinilai awalkan. Program ini akan memasukkan mod ganjil, semasa menunggu input pengguna dimasukkan. Jika pengguna memilih untuk mengubah nilai input, program akan mengubah output mengikut input pengguna. Jika pengguna menekan butang *start*, pengiraan warna langit akan bermula. Panel paparan akan menunjukkan simulasi bagi warna langit itu. Apabila pengguna beralih ke navigasi *trackbar*, kamera akan berputar samada mendatar atau menegak. Penjana dan paparan *group box* adalah opsiyen yang akan memberi pilihan kepada pengguna yang ingin menggunakan kaedah penjanaan dan paparan. Simulasi akan tamat sekiranya pengguna ingin keluar daripada program ini.

3.3.3.1 Aliran Aturcara



Rajah 3.1 : Aliran aturcara projek

3.3.3.2 Rekabentuk Objek Kelas

Perisian akan dibincangkan dalam pendekatan orientasi objek. Oleh itu, objek kelas dalam aturcara digambarkan sebagai rajah kelas objek. Di dalam perisian ini, terdapat lima kelas yang disediakan, di mana ia perlu bagi membangunkan program ini. Kelas hemisfera akan mengawal elemen yang berkaitan dengan rupa hemisfera langit. Ini termasuk membina dan menjana hemisfera. Kelas yang terlibat adalah kelas TskyColour, di mana ia akan mengira nilai warna bagi setiap titik. Kelas TskyColour juga akan mengawal pengiraan warna langit dengan menggunakan model Perez, di mana ia diperlukan sebagai nilai input untuk mengira kedudukan matahari. Nilai bagi warna langit akan diperoleh daripada kelas TsunPosition, Kelas CIE dan Tcolour akan mengawal warna CIE Yxy dan warna RGB.

3.3.3.3 Rekabentuk Antaramuka Pengguna

Bahagian ini akan membincangkan bagaimana sesuatu perisian akan dipaparkan kepada pengguna. Antaramuka pengguna adalah elemen yang utama bagi pembagunan perisian. Antaramuka direkabentuk menurut konsep kebolegunaan.

Kajian dalam kebolegunaan yang dikenali *Human Computer Interaction*. Kajian ini secara asasnya adalah berkenaan tentang cara yang berkesan untuk komputer mahupun perisian berinteraksi atau berkomunikasi dengan pengguna. Dalam projek ini, beberapa prinsip diambil daripada Hollingworth et al. (2001) akan diikuti untuk melaksanakan antaramuka pengguna. Antaranya ialah :

- Memenuhi kehendak pengguna
- Antaramuka ringkas dan mudah digunakan
- Terdapat keseragaman dalam antaramuka
- Antaramuka mudah difahami
- Antaramuka mesra pengguna
- Jalan keluar yang jelas
- membenarkan sebarang perubahan dilakukan
- Pengguna boleh menggunakan segala fungsi untuk memasukkan data

3.3.4 Fasa IV : Pengujian

Kefungsian dan ketepatan hasil output bagi perisian, dirancang untuk diuji bagi memastikan hasil perisian mengikut kehendak pengguna, suatu soal selidik dijalankan.

Untuk menguji kepersisan bagi pengiraan kedudukan matahari, ia dibandingkan dengan data sebenar. Masa sebenar bagi terbit, terbenamnya matahari akan dibandingkan dengan hasil pengiraan. Bagi memaparkan warna langit di dalam OpenGL, warna perlu dinyatakan dalam warna RGB. Dari sekarang, kiraan nilai warna bagi langit dalam warna CIE Yxy mesti ditukar kepada warna RGB. Bagi memastikan penukaran berjaya, perisian penukaran warna akan dibangunkan. Dengan ini, banyak penukaran matrik boleh diuji. Kemudian, matrik yang sesuai akan dipilih. Warna yang terhasil juga akan dibandingkan dengan perisian Adobe Photoshop, selain dunia sebenar yang diambil daripada kamera web.

3.3.5 Fasa V : Dokumentasi

3.3.5.1 Perancangan Masa

Sumber yang dianggap penting adalah masa. Projek dianggap berjaya sekiranya mengikut masa yang ditetapkan. Secara asasnya, projek akan dibahagikan kepada empat peringkat utama.

Peringkat 1 : Ulasan Literatur dan peringkat matematik

- Pengiraan warna langit
 - carian algoritma warna langit yang terbaik (carian pustaka)
 - studi serakan cahaya (1 minggu)
 - studi model analitik (1 minggu)
- Analisis dan Rekabentuk perisian (1 minggu)

Peringkat 2 : Pengaturcaraan

- Pengiraan kedudukan matahari (1 minggu)
- Permodelan hemisfera langit (1 minggu)
- Penukaran warna (1 minggu)
- Warna langit (2 minggu)

Peringkat 3 : Rekabentuk antaramuka pengguna dan dokumentasi

- Rekabentuk antaramuka pengguna bergrafik (1 minggu)
- Menulis laporan (3 minggu)

Peringkat 4 : Pengujian dan penilaian

- Pengujian perisian (2 hari)
- Penilaian terhadap hasil output (dibandingkan dengan data sebenar) (5 hari)

3.4 Analisa Keperluan Pembangunan

Bagi melaksanakan projek ini, spesifikasi alatan berbentuk perkakasan dan perisian diperlukan. Cadangan bagi keperluan perkakasan adalah seperti berikut :

3.4.1 Keperluan Perkakasan secara minimum :

1. Pemproses Intel Celeron 333 MHz ke atas.
2. Saiz ingatan sekurang-kurangnya 64MB.
3. Cakera keras dengan 10 GB.
4. Pemacu cakera liut 3 ¼"
5. CD Rom dengan kelajuan 50X
6. Kad SVGA yang boleh menampung OpenGL sekurang – kurangnya 2 MB memori video
7. Sistem pengoperasian WindowsNT 4.0, Windows 98, Windows ME dan Windows 2000

8. Kad Paparan Grafik Nvidia Gforce MX 32 MB
9. Monitor 17"
10. Tetikus
11. Papan kekunci.

3.4.2 Keperluan Perisian :

Cadangan bagi keperluan perisian adalah seperti berikut :

1. Java 2 SDK versi 1.3 atau terkini
2. JBuilder 5.
3. Java 3D(TM) SDK untuk tujuan pembangunan atau Java 3D runtime environment untuk tujuan melarikan aturcara.
4. Microsoft Visual C++ 6.0. Projek akan dijana mengikut pendekatan orientasi objek
5. Borland C++ Builder 5.0
6. OpenGL versi 1.2. OpenGL diperlukan dalam pengaturcaraan grafik bagi memaparkan hasil output.
7. Komponen OpenGLDP versi 1.0 untuk Borland C++ Builder 5.0. Ia digunakan bagi mengawal pustaka OpenGL dalam Borland C++ Builder, selain membantu dalam persekitaran Window.
8. 3D Studio Max versi 4.0 . Ia sebagai pilihan. 3D Studio Max adalah satu kaedah untuk memodelkan hemisfera.

3.5 Penutup

Dalam bab ini diterangkan mengenai penghasilan model langit, algoritma yang digunakan untuk menghasilkan warna langit, penggunaan pustaka grafik yang bersesuaian dan rekabentuk antaramuka pengguna untuk tujuan perbandingan. Di sini diterangkan secara jelas bagaimana projek ini akan dijalankan, algoritma untuk menghasilkan permodelan hemisfera, pengiraan warna langit dan penukaran ke dalam kategori warna RGB. Semua persamaan–persamaan untuk menghasilkan pengiraan di atas ditunjukkan.

BAB 4

IMPLEMENTASI

4.1 Pendahuluan

Di dalam bab ini, diterangkan mengenai kaedah Preetham yang berdasarkan kepada model Perez. Segala proses pengiraan untuk mendapatkan warna langit akan diperjelaskan di sini. Susunan algoritma bagi setiap langkah akan di huraikan. Langkah-langkah yang diterangkan di sini merupakan langkah-langkah yang dihasilkan di dalam aturcara.

4.2 Penghasilan Kedudukan Matahari

Kedudukan cahaya matahari memainkan peranan yang penting dalam menentukan warna pada langit. Kedudukan matahari yang berlainan memberikan kesan kepada warna langit.

Kedudukan matahari bergantung kepada sudut zenith (θ) dan azimuth (ϕ) di mana ianya bergantung kepada masa, tarikh, latitud dan longitud. Sudut zenith adalah sudut yang mewakili kedudukan matahari naik, iaitu kedudukan dari kaki langit hingga tegak 90 darjah ke atas. Manakala sudut azimuth adalah sudut yang mengelilingi hemisfera berkenaan.

Untuk mengira masa piawai, formula di bawah digunakan,

$$t = t_s + 0.170 \sin\left(\frac{4\pi(J-80)}{373}\right) - 0.129 \sin\left(\frac{2\pi(J-8)}{355}\right) + 12\left(\frac{SM-L}{\pi}\right) \dots (4.1)$$

di mana t adalah masa di dalam jam, t_s adalah masa piawai di dalam jam, SM adalah masa piawai meridian untuk zon masa didalam radians dan L adalah longitud di dalam radian. Nilai bagi solar declination dianggarkan seperti berikut:

$$\delta = 0.4093 \sin\left(\frac{2\pi(J-81)}{368}\right) \dots \dots \dots (4.2)$$

di mana δ adalah solar declination dalam radians, J adalah hari Julian (iaitu hari dalam tahun di ambil dengan nilai integer dari 1 hingga 365). Oleh yang demikian, kedudukan matahari (θ_s, ϕ_s) berdasarkan kepada solar declination, latitud dan longitud adalah seperti berikut:

$$\theta_s = \frac{\pi}{2} - \arcsin (\sin l \sin \delta - \cos l \cos \delta \cos \frac{\lambda t}{12}) \dots\dots\dots(4.3)$$

$$\phi_s = \arctan \left(\frac{-\cos \delta \sin \frac{\lambda t}{12}}{\cos l \sin \delta - \sin l \cos \delta \cos \frac{\lambda t}{12}} \right), \dots\dots\dots(4.4)$$

di mana θ_s adalah darjah zenith dalam radians, ϕ_s darjah azimuth dalam radians, l adalah latitud dalam radians, δ adalah solar declination dalam radians dan t adalah masa di dalam unit desimal. Nilai zenith adalah diantara 0 hingga $\pi/2$. Nilai yang lebih daripada $\pi/2$ adalah kedudukan matahari di bawah kaki langit. Dengan pengiraan kedudukan matahari secara persis, warna langit dapat dikira daripada nilai θ_s dan ϕ_s . Algoritma untuk pengiraan kedudukan matahari ditunjukkan di dalam Rajah 4.0.

```

double solarTime;
double solarAltitude;
double solarAzimuth;
double solarDeclination;
double opp,adj;

solarTime = standardtime +
    0.170 * Math.sin(4 * Math.PI * (JulianDay - 80) /
    373) - 0.129 * Math.sin(2 * Math.PI * (JulianDay - 8) /
    355)) + 12 * (standardMeridian - longitude) / Math.PI;

solarDeclination = (0.4093 * Math.sin(2 * Math.PI * (JulianDay -
    81) / 368));

solarAltitude = Math.asin(Math.sin(Math.toRadians(latitude)) *
    Math.sin(solarDeclination) -
    Math.cos(Math.toRadians(latitude)) *
    Math.cos(solarDeclination) *
    Math.cos(Math.PI * solarTime / 12));

opp = -Math.cos(solarDeclination) * Math.sin(Math.PI *
    solarTime / 12);
adj = -(Math.cos(Math.toRadians(latitude)) *
    Math.sin(solarDeclination) +
    Math.sin(Math.toRadians(latitude)) *
    Math.cos(solarDeclination) *
    Math.cos(Math.PI * solarTime / 12));

solarAzimuth = Math.atan2(opp, adj);

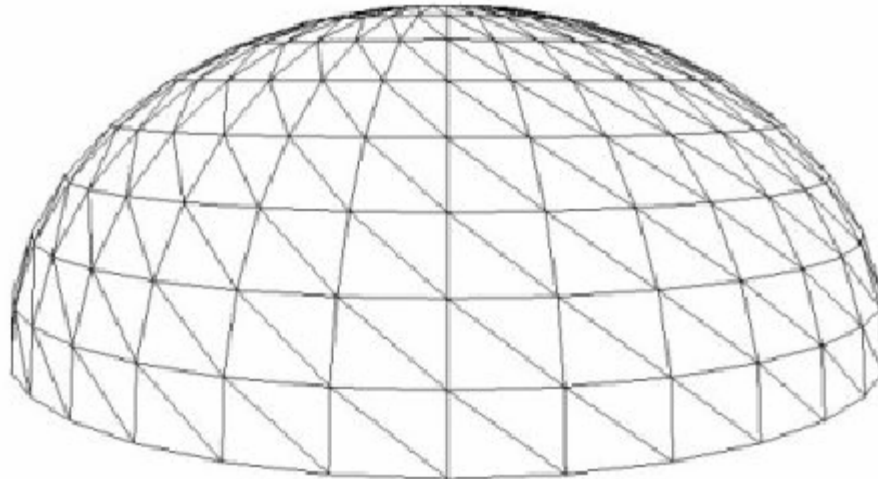
phiS = -solarAzimuth;
thetaS = Math.PI / 2.0 - solarAltitude;

```

Rajah 1.0 : Algoritma menentukan kedudukan matahari

4.3 Penghasilan Model Langit Hemisfera

Model hemisfera digunakan sebagai model bagi langit. Model hemisfera dapat menggambarkan keadaan langit yang sebenar di mana pandangan pemerhati akan diletakkan di dalam hemisfera yang dihasilkan. Warna yang dihasilkan nanti akan diletakan pada hemisfera. Gambaran hemisfera yang akan dihasilkan adalah seperti Rajah 4.1



Rajah 4.1 : Model hemisfera

Terdapat pelbagai cara untuk memodelkan hemisfera. Antaranya dengan menggunakan 3D Studio Max. Kemudian model ini akan dihantar sebagai format fail ASE (ASCII Scene Export) kepada program yang ingin dibuat. Kaedah lain adalah dengan menggunakan fungsi matematik, iaitu dengan menggunakan pengiraan *spherical*.

Namun *parser* diperlukan untuk mengeksport data. Oleh itu, format fail ASE perlu difahami terlebih dahulu. Terdapat beberapa kelemahan menggunakan teknik ini. Data yang dihasilkan oleh 3D Studio Max sukar diuruskan. Bilangan bagi titik-titik bergantung kepada permodelan hemisfera yang dibina. Ia mudah berubah bagi setiap model hemisfera. Begitu juga untuk membuat sebarang perubahan kepada model, perubahan juga perlu dibuat ke atas model utama di dalam 3D Studio Max *modeller*. Di samping, ia memerlukan ruang storan yang banyak bagi menyimpan memori dan fail data.

Oleh sebab itu, pendekatan matematik dipilih untuk memodelkan hemisfera. Titik-titik dikira dengan menggunakan pengiraan *sphere* yang standard. Sudut zenith

dan sudut azimuth bagi hemisfera ditunjukkan dalam Rajah 4.2. Di bawah fungsi asas bagi *spherical*, koordinat bagi titik-titik dalam hemisfera diperincikan dengan x,y,z iaitu:

$$x = \cos \phi \cdot \cos \theta$$

$$y = \sin \theta$$

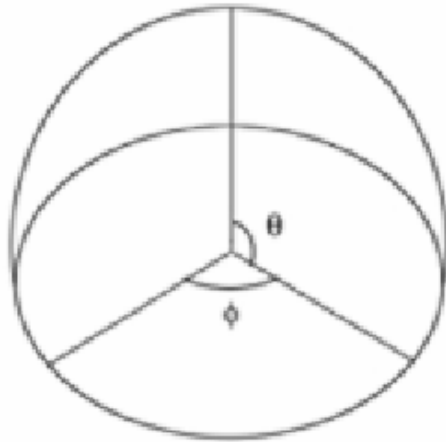
$$z = \cos \phi \cdot \sin \theta$$

di mana, θ adalah sudut zenith (satah menegak) dan ϕ ialah sudut azimuth (satah mendatar). Nilai θ dan ϕ diperincikan dalam radians dan julatnya adalah antara :

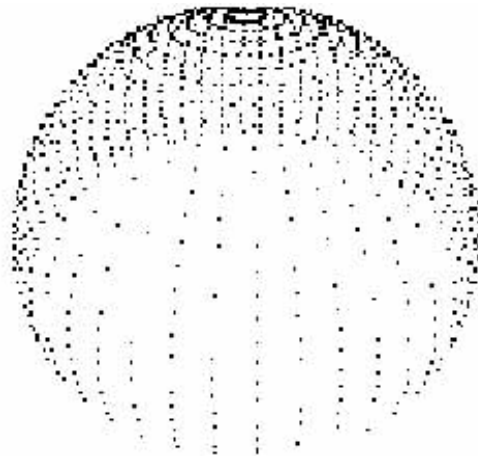
$$0 \leq \theta \leq \pi/2 .$$

$$0 \leq \phi \leq 2\pi$$

Titik– titik pada model hemisfera disimpan dalam tatasusunan 2D dan bilangan pada titik – titik boleh ditambah dengan meningkatkan nilai kejituan. Selain, titik tengah dan radians bagi hemisfera juga perlu diperincikan. Rajah 4.3 menunjukkan titik-titik yang dihasilkan oleh model hemisfera.



Rajah 4.2 : Gambaran sudut bagi zenith θ dan azimuth ϕ



Rajah 4.3 : Gambaran titik – titik yang di hasilkan

Dalam pelaksanaan, kelas hemisfera dibina untuk mengawal perkaitan tentang hemisfera. Bagi memaparkan hemisfera dalam OpenGL, terdapat beberapa langkah yang perlu diikuti. Jika hemisfera dipaparkan dalam mod pembayang, atribut `GL_SMOOTH` dan `GL_TRIANGLE_STRIP` diaplikasikan. Namun, jika hemisfera ingin dipaparkan sebagai *wireframe* atribut di atas diganti dengan `GL_LINE_STRIP`. Namun untuk mendapat hasil yang baik, *tringle strip* digunakan sebagai tujuan penjanaan.

Walau bagaimanapun, pengiraan bagi fungsi matematik perlu dimodifikasikan sedikit memandangkan fungsi OpenGL tidak boleh dilaksanakan secara terus. Berikut merupakan algoritma bagi modifikasi ini :

```
Void dome :: createDome (const int n, const double theta1, const double theta2, const
double phi1, const double phi2) {
```

```
    int i,j;
    double t1,t2,t3,x,y,z;
    //normal vector
    Tvector e;

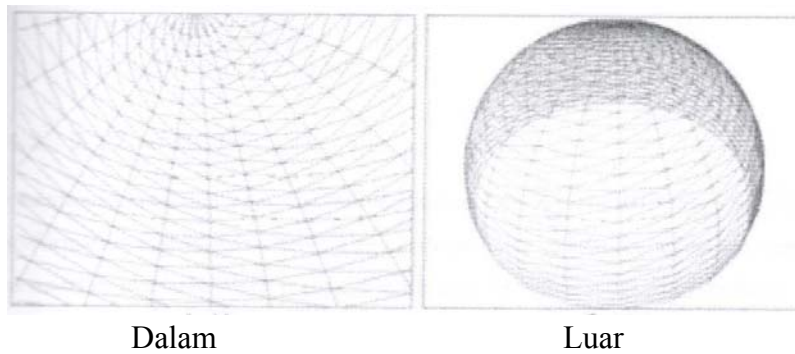
    //loop to fit with the triangular mesh
    //x = cos phi * cos theta
    //y= sin theta

    //z= cos phi * sin theta
    for (j=0; j<n/2; j++) {
        t1 = phi1 + j * ( phi2 - phi1 ) / ( n/2 );
        t2 = phi1 + ( j + 1 ) * ( phi2 - phi1 ) / ( n/2 );
        for (i=0; i<n*2; i++) {
            t3 = theta1 + 1 * ( theta2 - theta1 ) / n;
            x = cos ( t1 ) * cos ( t3 );
            e.setX( x );
            y = sin ( t1 );
            e.setY( y );
            z = cos( t1 ) * sin ( t3 );
```

```

e.setZ ( z );
//set normal vector
normal[j][i].setVector ( e );
//calculate the vertex (x,y,z)
x = center.X() + radius * e.X();
vertex[j][i].setX(x);
y = center.Y() + radius * e.Y();
vertex[j][i].setY(y);
z = center.Z() + radius * e.Z();
vertex[j][i].setZ(z);
i++;
//for neighbouring vertex
x = cos ( t2 ) * cos ( t3 );
e.setX(x);
y = sin ( t2 );
e.setY(y);
z = cos ( t2 ) * sin ( t3 );
e.setZ(z);
//set normal vector
normal[j][i].setVector(e);
calculate the vertex (x,y,z)
x = center.X() + radius * e.X();
vertex[j][i].setX(x);
y = center.Y() + radius * e.Y();
vertex[j][i].setY(y);
z = center.Z() + radius * e.Z();
vertex[j][i].setZ(z);
}
}
}

```



Rajah 4.4 : Keadaan dalam dan luar *wireframe* bagi hemisfera langit

Rajah 4.4 menunjukkan pelaksanaan hemisfera menggunakan *wireframe*. Ditunjukkan bahawa pandangan untuk pemerhati atau kamera diletakkan pada kedua-duanya dalam dan luar hemisfera.

Setiap titik pada hemisfera akan mempunyai nilai warna tersendiri. Gabungan bagi keseluruhan warna akan menghasilkan warna langit.

Selain pendekatan matematik di atas, terdapat pendekatan yang lain. Untuk menghasilkan satu sfera, persamaan untuk menghasilkannya adalah:

$$x^2 + y^2 + z^2 = r^2 \quad \dots\dots\dots(4.5)$$

Sfera yang berkedudukan pada origin dalam koordinat Cartesian dengan jejari r digambarkan dengan fungsi berikut:

$$f(p) = x^2 + y^2 + z^2 - r^2 = 0 \quad \dots\dots\dots(4.6)$$

di mana p adalah setiap titik pada sfera. Untuk mendapatkan setiap titik pada sfera, maka setiap titik diberi oleh persamaan berikut.

$$p = (px, py, pz) = f(\theta, \phi) = (fx(\theta, \phi), fy(\theta, \phi), fz(\theta, \phi)) \quad \dots(4.7)$$

di mana θ sudut bagi zenith, dan ϕ sudut bagi azimuth. Gambaran sudut seperti Rajah 4.2

Setiap titik bagi x , y dan z adalah diwakili persamaan berikut:

$$\begin{aligned} fx(\theta, \phi) &= r \sin(\theta) \cos(\phi) \\ fy(\theta, \phi) &= r \sin(\theta) \sin(\phi) \\ fz(\theta, \phi) &= r \cos(\theta) \quad \dots\dots\dots(4.8) \end{aligned}$$

Untuk menghasilkan sfera sudut bagi θ di antara -90 darjah ($-\pi/2$) hingga 90 darjah ($\pi/2$) dan ϕ diantara 0 hingga 360 darjah. Bagi menghasilkan model hemisfera iaitu separuh daripada sfera, sudut bagi θ diantara 0 darjah hingga 90 darjah ($\pi/2$) dan ϕ diantara 0 hingga 360 darjah.

Daripada persamaan ini, maka dapat dihasilkan titik-titik pada hemisfera. Gambaran hasil janaan titik seperti ditunjukkan di dalam Rajah 4.3

Untuk menghasilkan model hemisfera, maka perlu disambungkan ke semua titik–titik tersebut untuk dijadikan satu siri segitiga–segitiga. Untuk menghasilkan satu siri segitiga–segitiga, 4 titik diperlukan di mana 3 titik digunakan untuk menghasilkan sebuah segitiga dan titik keempat digunakan untuk menghasilkan segitiga kedua dengan menggunakan 2 titik sebelumnya. Titik–titik yang diperlukan untuk menghasilkan adalah:

1. $f(\theta, \phi)$
2. $f(\theta + \Delta\theta, \phi)$
3. $f(\theta, \phi + \Delta\phi)$
4. $f(\theta + \Delta\theta, \phi + \Delta\phi)$ (4.9)

Perlu ditentukan terlebih dahulu berapa banyak titik yang diperlukan untuk menghasilkan hemisfera dengan membahagikan 360 dengan $\Delta\phi$ dan 90 dengan $\Delta\theta$, kemudian kedua–dua hasil didarabkan:

$$\text{Jumlah titik pada hemisfera} = (360 / \Delta\phi) * (90 / \Delta\theta) \dots\dots\dots(4.10)$$

Memandangkan setiap siri segitiga–segitiga dihubungkan dengan 4 titik, maka perlu didarabkan jumlah titik dengan 4 untuk mendapatkan jumlah titik yang sebenar:

$$\text{Jumlah titik pada hemisfera} = (360 / \Delta\phi) * (90 / \Delta\theta) * 4 \dots\dots\dots(4.11)$$

Algoritma penghasilan model hemisfera ditunjukkan di dalam Rajah 4.5

```

Point3f[] Vertices;
int NumVertices;

public SkyDome(float radius, float dtheta, float dphi)
{
    int theta, phi;
    int n = 0;
    float x = 0;
    float y = 0;
    float z = 0;

    NumVertices = (int)((360/dtheta)*(90/dphi)*4);
    Vertices = new Point3f[NumVertices];
    Theta = new double[NumVertices];
    Phi = new double[NumVertices];

    for (theta=0; theta <= 90 - dtheta; theta += (int)dtheta)
    {
        for (phi=0; phi <= 360 - dphi; phi += (int)dphi)
        {
            x = radius * (float)(Math.sin(Math.toRadians(theta))) *
                (float)(Math.cos(Math.toRadians(phi)));
            y = radius * (float)(Math.sin(Math.toRadians(theta))) *
                (float)(Math.sin(Math.toRadians(phi)));

            z = radius * (float)(Math.cos(Math.toRadians(theta)));

            Vertices[n] = new Point3f(x,y,z);
            Phi[n] = Math.toRadians(phi);
            Theta[n] = Math.toRadians(theta);
            n++;

            x = radius * (float)(Math.sin(Math.toRadians(theta+dtheta))) *
                (float)(Math.cos(Math.toRadians(phi)));
            y = radius * (float)(Math.sin(Math.toRadians(theta+dtheta)))
                *(float)(Math.sin(Math.toRadians(phi)));
            z = radius * (float)(Math.cos(Math.toRadians(theta+dtheta)));
        }
    }
}

```

Rajah 4.2 : Algoritma menghasilkan hemisfera


```

Vertices[n] = new Point3f(x,y,z);
Phi[n] = Math.toRadians(phi);
Theta[n]=Math.toRadians(theta+ dtheta);
n++;

    x = radius * (float)(Math.sin(Math.toRadians(theta))) *
              (float)(Math.cos(Math.toRadians(phi+ dphi)));

    y = radius * (float)(Math.sin(Math.toRadians(theta))) *
      (float)(Math.sin(Math.toRadians(phi+dphi)));
    z = radius * (float)(Math.cos(Math.toRadians(theta)));

    Vertices[n] = new Point3f(x,y,z);
    Phi[n]=Math.toRadians(phi+ dphi);
    Theta[n]=Math.toRadians(theta);
    n++;

if (theta > -90 && theta < 90)
{
    x = radius * (float)(Math.sin(Math.toRadians(theta+dtheta))) *
              (float)(Math.cos(Math.toRadians(phi+ dphi)));

    y = radius * (float)(Math.sin(Math.toRadians(theta+dtheta))) *
      (float)(Math.sin(Math.toRadians(phi+dphi)));

    z = radius * (float)(Math.cos(Math.toRadians(theta+ dtheta)));

    Vertices[n] = new Point3f(x,y,z);
    Phi[n]=Math.toRadians(phi+ dphi);
    Theta[n]=Math.toRadians(theta+ dtheta);
    n++;
}
}
}
}
}

```

Rajah 4.5 : (Sambungan) Algoritma menghasilkan hemisfera

4.4 Penukaran Nilai Warna

Pengiraan bagi warna langit menggunakan model Perez dalam Perez et al. (1993), akan menghantarkan nilai koefisien dan zenith untuk *luminance* Y, dan *chromaticities* x dan y. Oleh itu, nilai dalam ruang warna Yxy dalam pengiraan perlu ditukarkan kepada ruang warna RGB. Ini kerana OpenGL API¹ hanya dapat menyokong ruang warna RGB dalam pemaparan warna. Untuk penukaran ruang warna daripada CIE Yxy kepada RGB, CIE Yxy perlu ditranslasikan kepada ruang warna CIE XYZ sebelum kepada ruang warna RGB.

4.4.1 Yxy kepada XYZ

Penukaran warna daripada Yxy kepada XYZ dengan menggunakan persamaan berikut:

$$\begin{aligned}
 Y &= Y \\
 X &= (x * y) * Y \\
 Z &= ((1 - x - y) / y) * Y \quad \dots\dots\dots(4.12)
 \end{aligned}$$

¹ Application Programming Interface

4.4.2 XYZ kepada RGB

Penukaran daripada warna XYZ kepada RGB tidaklah begitu mudah. Matrik transformasi yang digunakan bergantung kepada jenis monitor yang digunakan. Terdapat banyak jenis matrik transformasi yang ada dan setiap jenis matrik bergantung kepada jenis monitor yang berlainan.

Matrik yang telah diperkenalkan oleh Watt telah digunakan kerana ianya memberikan warna yang lebih hampir kepada warna RGB yang dikehendaki.

Untuk mendapatkan matrik penukaran adalah seperti dibawah:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = T \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots\dots\dots(4.13)$$

di mana

$$T = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \dots\dots\dots(4.14)$$

T adalah matrik transformasi untuk penukaran CIE kepada RGB. Matrik T bergantung kepada jenis monitor yang digunakan. Sekiranya T1 digunakan untuk monitor 1 dan T2 digunakan untuk monitor 2, maka proses penukaran RGB daripada

monitor 1 kepada monitor 2 adalah dengan matrik T2-1 T1. Nilai bagi T dikira seperti berikut:

Katakan D_r , D_g D_b

$$D_r = X_r + Y_r + Z_r$$

$$D_g = X_g + Y_g + Z_g$$

$$D_b = X_b + Y_b + Z_b \quad \dots\dots\dots (4.15)$$

Diberikan :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} D_r x_r & D_g x_g & D_b x_b \\ D_r y_r & D_g y_g & D_b y_b \\ D_r z_r & D_g z_g & D_b z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \dots\dots\dots(4.16)$$

dimana :

$$x_r = \frac{X_r}{D_r}, y_r = \frac{Y_r}{D_r}, z_r = \frac{Z_r}{D_r} \quad \dots\dots\dots(4.17)$$

Ianya dapat menghasilkan 2 matrik yang berasingan iaitu:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} D_r & 0 & 0 \\ 0 & D_g & 0 \\ 0 & 0 & D_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \dots\dots\dots(4.18)$$

Matrik yang pertama mewakili nilai koordinat pada fosfor di monitor. Katakan nilai RGB adalah (1,1,1) dimana ianya akan menghasilkan warna putih:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} D_r \\ D_g \\ D_b \end{bmatrix} \quad \dots\dots\dots(4.19)$$

Sebagai contoh untuk warna putih D65;

$$x_w = 0.313 \quad y_w = 0.329 \quad z_w = 0.358 \quad \dots\dots(4.20)$$

dan penskalaan untuk mendapatkan warna putih

$$X_w = 0.951 \quad Y_w = 1.0 \quad Z_w = 1.089 \quad \dots\dots(4.21)$$

Contoh jadual koordinat bagi monitor .

Jadual 4.0 : Jadual koordinat bagi monitor

	x	y
Merah	0.620	0.330
Hijau	0.210	0.685
Biru	0.150	0.063

Dengan menggunakan jadual diatas,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.584 & 0.188 & 0.179 \\ 0.311 & 0.614 & 0.075 \\ 0.047 & 0.103 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \dots\dots(4.22)$$

dengan menggunakan matrik songsang akan menghasilkan nilai RGB.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.403 & -0.568 & -0.344 \\ -1.036 & 1.939 & 0.043 \\ 0.011 & -0.184 & 1.078 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \dots\dots(4.23)$$

```
Y = Ys;  
X = (x / y) * Y;  
Z = ((1-x-y) / y) * Y;  
  
r = 2.043*X - 0.568*Y - 0.344*Z;  
g = -1.036*X + 1.939*Y + 0.043*Z;  
b = 0.011*X - 0.184*Y + 1.078*Z;
```

Rajah 4.3 : Algoritma penukaran warna daripada CIE – Yxy kepada CIE – XYZ dan daripada CIE – XYZ kepada RGB

Kadang kala nilai penukaran ini akan memberikan nilai negatif. Seperti yang kita ketahui bahawa nilai negatif dalam RGB tidak ada. Oleh yang demikian satu proses penskalaan perlu dilakukan. Penskalaan yang dilakukan ditunjukkan di dalam Rajah 4.7.

```
double lum = Y/255;

if (r<b)
{
    if (r<b)
    {
        min = r;
    }
    else
        min = b;
}
else
{
    if (g<b)
    {
        min = g;
    }
    else
        min = b;
}

if (r<0 || g<0 || b<0)
{
    min = -min;
    r += min;
    g += min;
    b += min;
}

r/=max;
g/=max;
b/=max;

r*=lum;
g*=lum;
b*=lum;

if (r<0)r = 0;
if (g<0)g = 0;
if (b<0)b = 0;

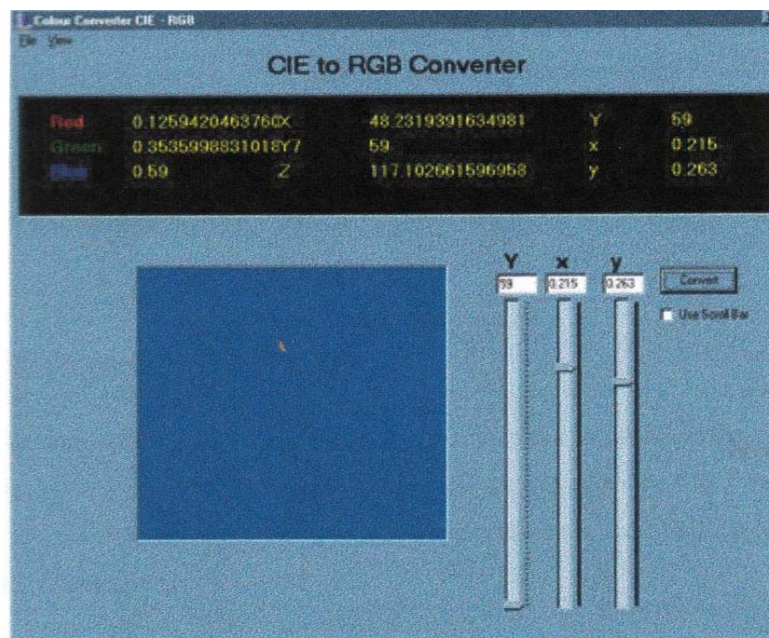
if (r>1)r = 1;
if (g>1)g = 1;
if (b>1)b = 1;

R = r;
G = g;
B = b;
```

Rajah 4.4 : Penskalaan untuk nilai RGB

4.4.3 Pembangunan Perisian Penukaran Khas Warna

Perisian penukaran warna dibuat semasa fasa pembangunan bagi projek ini. Tujuan ia dibuat adalah supaya pengiraan penukaran dibuat dengan betul. Selain, membantu dalam pemilihan matrik penukaran yang sesuai. Perisian juga boleh membantu dalam perbandingan hasil yang didapati. Rajah 4.8 menunjukkan *screen captured* bagi perisian penukaran ruang warna ini.



Rajah 4.5 : *Screen captured* bagi perisian penukaran ruang warna

4.5 Pengiraan Warna Langit

Bahagian ini akan mengambarkan langkah–langkah warna langit dihasilkan. Ia melibatkan pendekatan matematik, yang dihasilkan oleh Perez et al. (1993). Hasil yang membenarkan penghantaran nilai koefisien dan zenith kepada cahaya langit lebih dikenali sebagai model Perez.

4.5.1 Model Perez

Penggunaan model Perez memainkan peranan yang penting dalam menghasilkan warna pada langit. Beberapa keadaan perlu dipertimbangkan iaitu kedudukan matahari, masa, dan keadaan langit yang mempengaruhi kesan warna pada langit.

Di dalam model Perez, cahaya bagi (θ, ϕ) pada setiap titik pada hemisfera menggunakan fungsi:

$$L(\theta, \gamma) = (1 + Ae B / \cos \theta)(1 + CeD^\gamma + E \cos 2\gamma) \dots \dots (4.24)$$

Perez telah menghasilkan 5 parameter dalam modelnya untuk menggambarkan berkenaan dengan warna langit. Paramater – parameter ini diwakili dengan pembolehubah A, B, C, D dan E. Bagi setiap pembolehubah ini membawa maksud:

A = kegelapan atau kecerahan pada kaki langit.

B = kesan gradient pada horizon.

C = kecerahan lingkungan kawasan cahaya.

D = kelebaran lingkungan kawasan cahaya.

E = serakan cahaya yang berkaitan.

Nilai bagi gamma (γ) mewakili kedudukan sudut matahari dengan mana – mana titik pada hemisfera. θ_s dan ϕ_s mewakili posisi matahari. Nilai bagi γ diperolehi daripada :

$$\cos \gamma = (\sin \theta_s \cdot \sin \theta \cdot \cos (\phi - \phi_s)) \quad \dots\dots\dots(4.25)$$

Algoritma daripada persamaan 4.13 ditunjukkan di dalam Rajah 4.9

```

public double AngleInBetween(double theta, double phi)
{
    double cospsi;

    cospsi = Math.sin(theta) * Math.sin(thetaS) *
            Math.cos(phiS-phi) + Math.cos(theta) *
            Math.cos(thetaS);

    if(cospsi > 1.0) return 0;
    if(cospsi < -1.0) return Math.PI;

    return Math.acos(cospsi);
}

```

Rajah 4.6 : Algoritma mengira sudut antara matahari dengan mana –mana titik di dalam hemisfera

Cahaya bagi Y bagi langit diperolehi daripada fungsi yang telah diagihkan dengan menggunakan persamaan dibawah:

$$Y = Y_Z \frac{L(\theta, \gamma)}{L(0, \theta_s)} \quad \dots\dots\dots(4.26)$$

Kecerahan untuk fungsi bagi x dan y juga dapat dihasilkan dengan menggunakan persamaan dibawah:

$$y = yz \frac{L(\theta, \gamma)}{L(0, \theta_s)} \dots\dots\dots(4.27)$$

dan

$$x = xz \frac{L(\theta, \gamma)}{L(0, \theta_s)} \dots\dots\dots(4.28)$$

Parameter – parameter yang dihasilkan untuk kesan kecerahan bagi setiap komponen Y, x dan y adalah:

$$\begin{bmatrix} A_y \\ B_y \\ C_y \\ D_y \\ E_y \end{bmatrix} = \begin{bmatrix} 0.1787 & -1.4630 \\ -0.3554 & 0.4275 \\ -0.0227 & 5.3251 \\ 0.1206 & -2.5771 \\ -0.0670 & 0.3703 \end{bmatrix} \begin{bmatrix} T \\ 1 \end{bmatrix} \dots\dots\dots(4.29)$$

$$\begin{bmatrix} A_x \\ B_x \\ C_x \\ D_x \\ E_x \end{bmatrix} = \begin{bmatrix} -0.0193 & -0.2592 \\ -0.0665 & 0.0008 \\ -0.0004 & 0.2125 \\ -0.0641 & -0.8989 \\ -0.0033 & 0.0452 \end{bmatrix} \begin{bmatrix} T \\ 1 \end{bmatrix} \dots\dots\dots(4.30)$$

$$\begin{bmatrix} A_y \\ B_y \\ C_y \\ D_y \\ E_y \end{bmatrix} = \begin{bmatrix} -0.0167 & -0.2608 \\ -0.0950 & 0.0092 \\ -0.0079 & 0.2102 \\ -0.0441 & -1.6537 \\ -0.0109 & 0.0529 \end{bmatrix} \begin{bmatrix} T \\ 1 \end{bmatrix} \dots\dots\dots(4.31)$$

Kecerahan bagi sudut zenith Yz diperolehi dengan menggunakan formula untuk kedudukan cahaya matahari dan kesan kekeruhan pada langit. Persamaan untuk ketiga – tiga Yz, xz dan yz diperolehi seperti berikut:

Nilai mutlak bagi kecerahan pada zenith adalah seperti berikut

$$Y_z = (4.0453T - 4.9710) \tan \chi - 0.2155T + 2.4192 \dots\dots\dots(4.32)$$

Dimana nilai bagi $\chi = \left(\frac{4}{9} - \frac{T}{120}\right)(\pi - 2\theta_s)$ (4.33)

Nilai bagi zenith x ialah:

$$x_z = [T^2 \quad T \quad 1] \begin{bmatrix} 0.00166 & -0.00375 & 0.00209 & 0 \\ -0.02903 & 0.06377 & -0.03202 & 0.00394 \\ 0.11693 & -0.21196 & 0.06052 & 0.25886 \end{bmatrix} \begin{bmatrix} \theta_s^3 \\ \theta_s^2 \\ \theta_s \\ 1 \end{bmatrix} \dots\dots(4.34)$$

Nilai bagi zenith y ialah:

$$y_z = [T^2 \quad T \quad 1] \begin{bmatrix} 0.00275 & -0.00610 & 0.00317 & 0 \\ -0.04214 & 0.08970 & -0.04153 & 0.00516 \\ 0.15346 & -0.26756 & 0.06670 & 0.26688 \end{bmatrix} \begin{bmatrix} \theta_s^3 \\ \theta_s^2 \\ \theta_s \\ 1 \end{bmatrix} \dots\dots(4.35)$$

```

double chi;
double theta2 = thetaS * thetaS;
double theta3 = theta2 * thetaS;
double T = Turbidity;
double T2 = T * T;

chi = (4.0 / 9.0 - T / 120.0) * (Math.PI - 2 * thetaS);
Zenith_Y = (4.0453 * T - 4.9710) * Math.tan(chi) - 0.2155 * T + 2.4192;
Zenith_Y *= 1000;

Zenith_x =
  (+ 0.00165 * theta3 - 0.00374 * theta2 + 0.00208 * thetaS + 0) * T2 +
  (-0.02902 * theta3 + 0.06377 * theta2 - 0.03202 * thetaS + 0.00394) * T +
  (+ 0.11693 * theta3 - 0.21196 * theta2 + 0.06052 * thetaS + 0.25885);

Zenith_y =
  (+ 0.00275 * theta3 - 0.00610 * theta2 + 0.00316 * thetaS + 0) * T2 +
  (-0.04214 * theta3 + 0.08970 * theta2 - 0.04153 * thetaS + 0.00515) * T +
  (+ 0.15346 * theta3 - 0.26756 * theta2 + 0.06669 * thetaS + 0.26688);

perez_Y[0] = 0.17872 * T - 1.46303;
perez_Y[1] = -0.35540 * T + 0.42749;
perez_Y[2] = -0.02266 * T + 5.32505;
perez_Y[3] = 0.12064 * T - 2.57705;
perez_Y[5] = -0.06696 * T + 0.37027;

perez_x[0] = -0.01925 * T - 0.25922;
perez_x[1] = -0.06651 * T + 0.00081;
perez_x[2] = -0.00041 * T + 0.21247;
perez_x[3] = -0.06409 * T - 0.89887;
perez_x[4] = -0.00325 * T + 0.04517;

perez_y[0] = -0.01669 * T - 0.26078;
perez_y[1] = -0.09495 * T + 0.00921;
perez_y[2] = -0.00792 * T + 0.21023;
perez_y[3] = -0.04405 * T - 1.65369;
perez_y[4] = -0.01092 * T + 0.05291;

```

Rajah 4.7 : Algoritma menentukan nilai untuk zenith Y, x dan y serta parameter untuk A,B,C,D dan E

4.6 Antaramuka Pengguna Bergrafik

Perisian ini digunakan dalam persekitaran Window tidak kepada aplikasi konsol. Antaramuka bagi projek ini memodelkan warna langit yang dibangunkan menggunakan Borland C++ Builder Visual Component Library (VCL). VCL membenarkan pembinaan komponen Window seperti butang *trackbar* dan menu dengan menggunakan stail *drag* dan *drop*. Beberapa pengetahuan tentang konsep pengaturcaraan Window dan kebolegunaan yang dikenali sebagai *Human Computer Interaction* diperlukan untuk menghasilkan antaramuka yang baik. Tambahan pula, beberapa arahan yang spesifik dalam Borland C++ Builder juga diperlukan.

Untuk pengiraan kedudukan matahari, beberapa parameter diperlukan . Oleh sebab itu, sistem memerlukan input daripada pengguna bagi tujuan di atas. Dalam aplikasi konsol, sistem hanya boleh setkan nilai *default* bagi parameter atau mengambil input daripada pengguna sebelum proses penjanaan dilakukan. Parameter itu tidak boleh diubah semasa program berjalan. Ini adalah salah satu kelemahan aplikasi konsol. Sebaliknya, program yang menggunakan aplikasi Window mudah untuk mengambil input daripada pengguna.

Pengiraan kedudukan matahari memerlukan tarikh Julian sebagai input. Tarikh Julian berasaskan hari bagi 1 tahun, yang julatnya antara 1 hingga 365. Bagi memudahkan pengguna daripada mengalami kekeliruan dalam pengiraan, terdapat input yang membenarkan pengguna memasukkan data bagi tarikh untuk 1 tahun. Kemudian, sistem secara automatik akan menukarkan ia kepada tarikh Julian. Tambahan, lokasi bagi pemerhati iaitu longitud dan latitud sentiasa diberi nilai dalam darjah dan minit. Pengguna boleh memberi nilai longitud dan latitud tanpa memikirkan sebarang pertukaran yang perlu dibuat. Namun sistem memerlukan pengiraan dalam radius.

Begitu juga, pengguna boleh membuat navigasi di sepanjang skrin dengan menggunakan *trackbar* menegak dan mendatar yang disediakan. Pada dasarnya, rekabentuk ini digerakkan oleh orientasi kamera secara menegak dan mendatar. Algoritma di bawah dibina untuk menggerakkan orientasi kamera.

```
//horizontal camera orientation
void orientMe(float ang) {
    lx = sin (ang);
    lz = -cos (ang);
    glLoadIdentity( );
    OpenGL camera function
    gluLookAt(x,y,z,
              x + lx, y +ly, z + lz
              0.0f, 1.0f,0.0f);
}
```

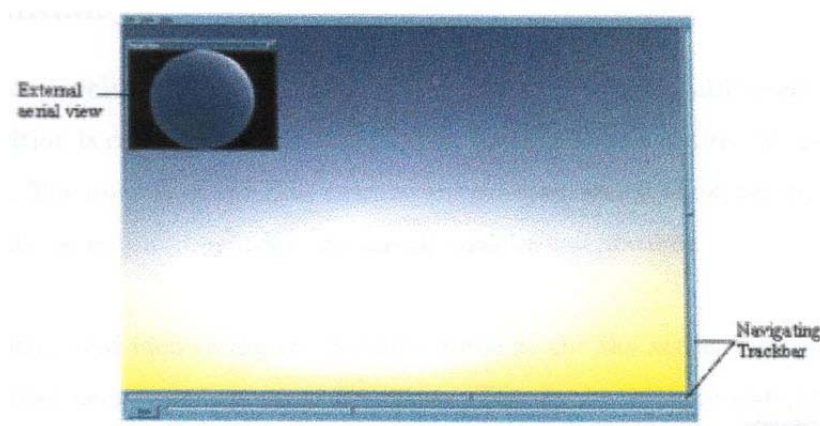
```
//Vertical camera orientation
void upDownMe (float ang) {
    ly = sin(ang);
    glLoadIdentity( );
    //OpenGL camera function
    gluLookAt (x, y, z,
              x + lx, y +ly, z + lz,
              0.0f, 1.0f,0.0f);
}
```

Pada mulanya, nilai *default* bagi sudut menegak dan mendatar disetkan kepada sifar. Apabila pengguna mengubah *trackbar*, sudut akan ditambah atau dikurangkan. Ini

akan memberi nilai input bagi fungsi `moveMe()` dan `upDownMe()`, bergantung pada *trackbar* yang dipilih oleh pengguna. Fungsi `moveMe()` dan `upDownMe()` akan mengubah parameter *look* bagi fungsi kamera OpenGL, `gluLookAt`.

Terdapat pelbagai cara untuk memaparkan program OpenGL. Pendekatan yang sering digunakan ialah menggunakan standard pustaka OpenGL Window bagi SGI (Silicon Graphics Inc.), GLUT². Namun terdapat kekompleksan apabila menggunakan GLUT dalam pengaturcaraan Window. Pendekatan alternatif adalah dengan menggunakan komponen OpenGLDP bagi Borland C++ Builder 5.0 sebagai menggantikan GLUT.

Dalam perisian ini, ia menyediakan skrin penuh. Namun pada mod ini, percubaan terhadap input data seperti lokasi dan tarikh tidak dibenarkan. Sistem akan disetkan ke mod normal apabila pengguna memilih *Normal* daripada menu view. Rajah 4.12 menunjukkan sistem berjalan dalam skrin penuh.



Rajah 4.8 : Mod skrin penuh bersama dengan pandangan aerial

² OpenGL Utility Toolkit

Pengguna juga boleh melihat perubahan warna langit daripada dunia luar menggunakan pandangan aerial. Ia disertakan pada tettingkap yang berbeza dengan tettingkap yang utama. Oleh itu, pengguna boleh melihat simulasi warna langit daripada permukaan bumi serentak dengan pandangan aerial. Sebenarnya, perubahan posisi pada kamera yang telah menghasilkan mod pandangan aerial, di mana kamera diletakkan di luar hemisfera langit.

Seperti yang dibincangkan sebelum ini, antaramuka pengguna direkabentuk mengikut konsep kebolegunaan. Antaramuka pengguna dibina supaya pengguna mudah untuk memasukkan input dan melihat perubahan warna langit berlaku dengan cepat. Perisian direka dengan Single Document Interface (SDI) untuk memudahkan pengguna menggunakannya serta mengelak daripada berlaku sebarang kekeliruan kepada pengguna. Penilaian terhadap antaramuka pengguna yang direka akan dibincangkan dalam Bab 5. Beberapa komen mengenai perisian ini juga disertakan.

4.7 Penutup

Di dalam bab ini, diterangkan secara lebih terperinci mengenai algoritma untuk menghasilkan permodelan hemisfera, pengiraan warna langit dan penukaran ke dalam kategori warna RGB. Semua persamaan – persamaan untuk menghasilkan pengiraan di atas juga ditunjukkan.

BAB 5

HASIL DAN PENGUJIAN

5.1 Pengenalan

Dalam bab ini, terdapat dua bahagian iaitu pengujian dan penilaian hasil. Fungsi perisian diuji supaya menepati kehendak pengguna. Beberapa orang pengguna diminta untuk menguji perisian ini. Kod aturcara juga diperiksa, bagi memastikan tidak berlaku pertindihan dengan pelbagai gabungan input yang dimasukkan oleh pengguna. Pengujian dilakukan ke atas aturcara untuk membandingkan keadaan langit dengan keadaan sebenar. Beberapa gambar di kawasan Universiti Teknologi Malaysia diambil untuk membuat perbandingan. Pengujian juga di lakukan untuk mengetahui keadaan waktu matahari terbit dan terbenam pada sesuatu kawasan. Selain daripada itu, pengujian ke atas pelayar web yang bersesuaian di lakukan untuk menguji kestabilan aturcara ke atas pelayar web yang berlainan.

Dalam aspek penilaian hasil pula, ketepatan hasil output diuji. Ia dilakukan secara langkah demi langkah. Pertama, pengiraan bagi kedudukan matahari diuji dengan menggunakan data sebenar bagi terbit dan terbenamnya matahari. Kedua, penukaran

matrik transformasi diuji supaya warna Yxy dapat memberikan hasil sebenar dalam warna RGB. Akhirnya, warna langit dalam perisian akan dibandingkan dengan pemandangan sebenar ditempat-tempat yang menarik di dunia yang diambil dengan menggunakan kamera web.

5.2 Hasil Kajian

5.2.1 Perbandingan Warna Langit

Imej yang dihasilkan oleh aturcara di bandingkan dengan keadaan sebenar. Untuk menentukan nilai persamaan antara keadaan langit dalam simulasi dengan keadaan sebenar, imej yang dihasilkan diambil dengan menggunakan kamera digital serta kamera web. Imej yang diambil dengan menggunakan kamera web dengan tujuan untuk membuat satu perbandingan. Terdapat banyak perkhidmatan kamera web di Internet.

Rajah 5.1 menunjukkan gambar Eiffel Tower di Paris yang diambil daripada France Telecom (2001) pada Selasa, 4hb September 2001, jam 7.40 malam. Manakala imej langit yang dihasilkan oleh perisian ditunjukkan pada Rajah 5.2. Koordinat yang diambil di Paris bagi latitud adalah $48^{\circ} 49'$ ke Utara, manakala bagi langitud pula adalah $2^{\circ} 29'$ ke Barat. Imej lain iaitu matahari terbenam diambil daripada kameraweb BBC – Online (2001). Pemandangan pantai di Bridlington, England yang lokasinya adalah $54^{\circ} 5'$ ke Utara dan $0^{\circ} 16'$ ke Barat diambil. Imej diambil pada Selasa, 4hb September 2001, jam 6.40 petang seperti yang ditunjukkan dalam rajah 5.3. Imej yang diambil boleh disimulasikan dalam perisian pada lokasi, masa dan tarikh yang sama seperti Rajah 5.4.

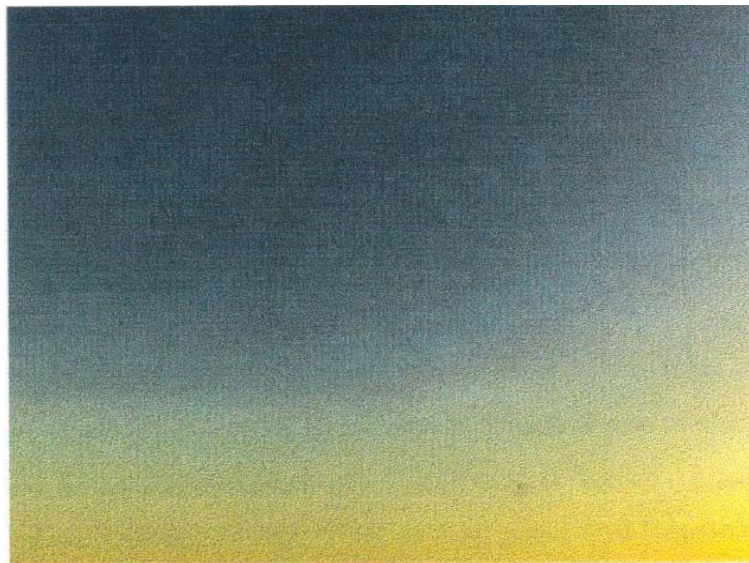
Bagi kamera digital pula, gambar sekitar Universiti Teknologi Malaysia diambil dengan menggunakan kamera digital. Gambar langit yang diambil merupakan gambar langit yang tidak berawan. Rajah 5.5 hingga 5.14 menunjukkan perbandingan antara warna langit yang dihasilkan di dalam projek ini dengan keadaan sebenar.

Oleh kerana keadaan langit yang ingin dibandingkan hanyalah keadaan langit yang tidak berawan, maka terlalu sukar untuk mendapatkan keadaan langit yang sebenar pada waktu-waktu yang dikehendaki. Keadaan langit berubah daripada tidak berawan pada waktu pagi dan berawan pada waktu petang.

Memandangkan keadaan warna langit hanya berubah berpandukan kepada kedudukan matahari, maka simulasi ini hanya berjaya untuk menunjukkan waktu matahari terbenam dan terbit pada satu-satu kawasan.



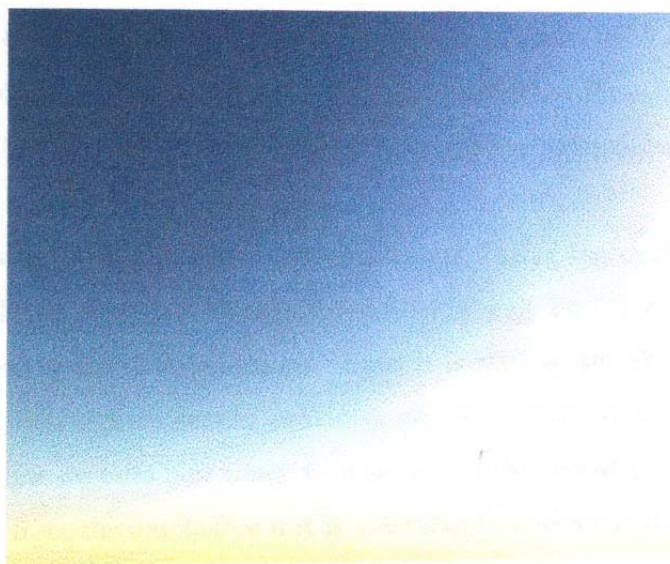
Rajah 5.1 : Eiffel Tower di Paris diambil pada 4 September 2001 jam 1940



Rajah 5.2 : Warna langit yang dijana untuk Eiffel Tower di Paris pada 4 September 2001 jam 1940



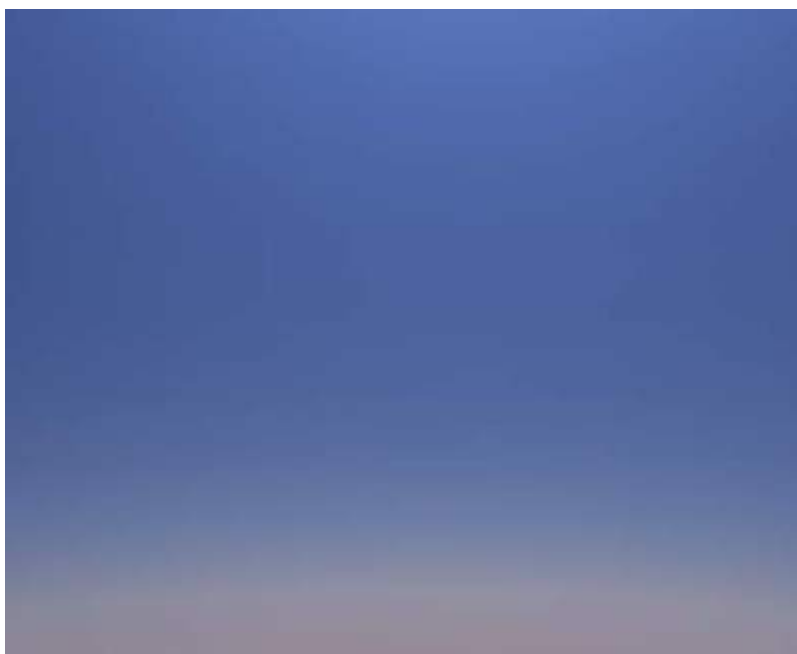
Rajah 5.3 : Pantai Bridlington, England pada 4 September 2001 jam 1840



Rajah 5.4 : Warna langit dijana untuk pantai Bridlington, England pada 4 September jam 1840



Rajah 5.5 : Gambar di kawasan Universiti Teknologi Malaysia pada 28 Julai 2002 jam 0845



Rajah 5.6 : Gambar yang diambil daripada aturcara yang dihasilkan pada 28 Julai 2002 jam 0845



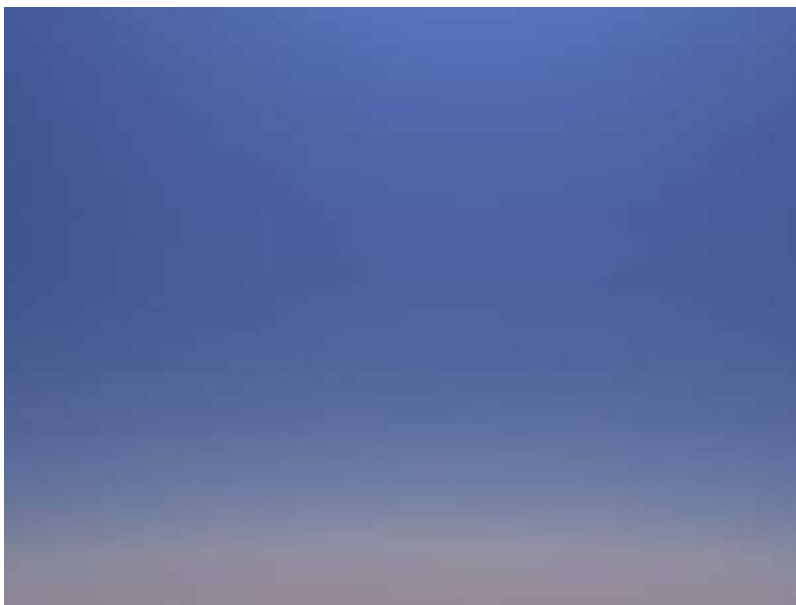
Rajah 5.7 : Gambar di sekitar kawasan Universiti Teknologi Malaysia pada 28 Julai 2002 jam 0830



Rajah 5.8 : Gambar yang diambil daripada aturcara yang dihasilkan pada 28 Julai 2002 jam 0830



Rajah 5.9 : Gambar di sekitar kawasan Universiti Teknologi Malaysia pada 28 Julai 2002 jam 0900



Rajah 5.10 : Gambar yang diambil daripada aturcara yang dihasilkan pada 28 Julai 2002 jam 0900



Rajah 5.11 : Gambar di sekitar kawasan Universiti Teknologi Malaysia pada 15 Ogos 2002 jam 1600



Rajah 5.1 : Gambar yang diambil daripada aturcara yang dihasilkan pada 15 Ogos 2002 jam 1600



Rajah 5.2 : Gambar di sekitar kawasan Desa Skudai pada 17 Ogos 2002 jam 1500

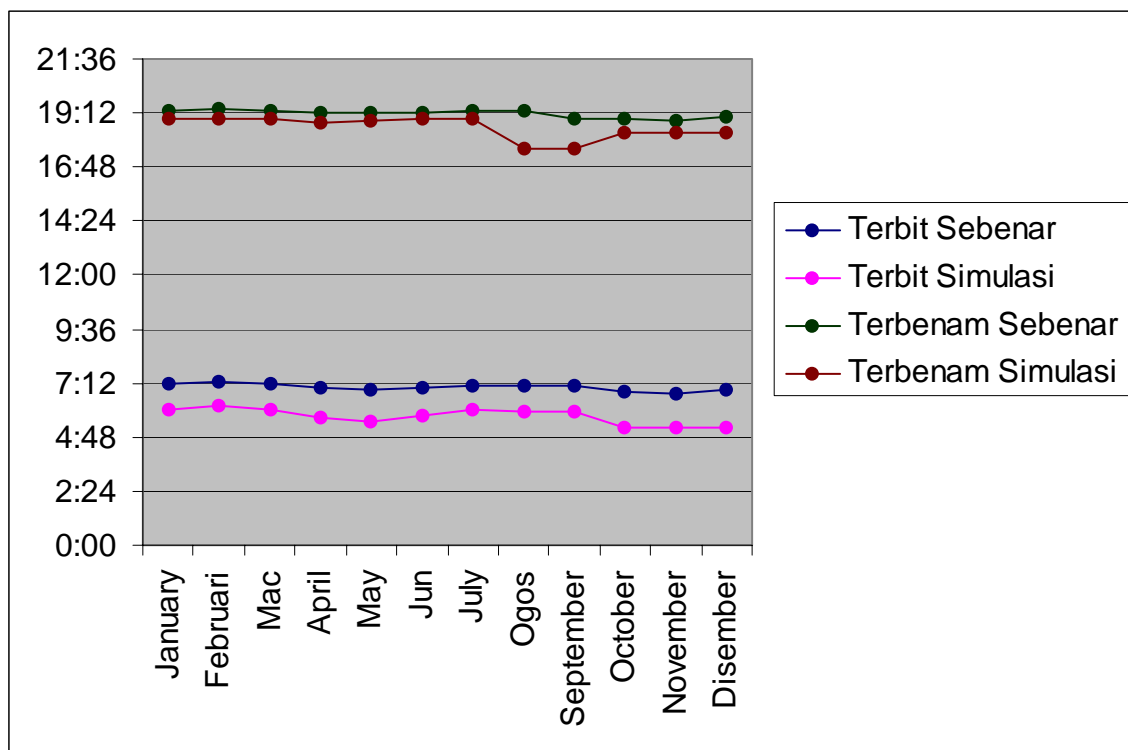


Rajah 5.3 : Gambar yang diambil daripada aturcara yang dihasilkan pada 17 Ogos 2002 jam 1500

Waktu mahari terbit dan terbenam

Jadual 5.2 : Jadual perbandingan antara matahari terbit dan terbenam

Matahari / Bulan	Terbit Sebenar	Terbit Simulasi	Terbenam Sebenar	Terbenam Simulasi
January	07:11	06:00	19:16	18:59
Februari	07:15	06:10	19:22	18:59
Mac	07:09	06:00	19:18	18:56
April	06:59	05:40	19:11	18:45
May	06:54	05:30	19:13	18:49
Jun	06:57	05:45	19:13	18:59
July	07:03	05:59	19:19	18:58
Ogos	07:03	05:55	19:16	17:35
September	07:05	05:55	18:55	17:35
October	06:47	05:15	18:55	18:18
November	06:46	05:15	18:52	18:20
Disember	06:56	05:15	19:01	18:22



Rajah 5.4 : Graf perbandingan di antara matahari terbit dan terbenam

Sebagaimana yang kita lihat, kedua-dua imej yang dihasilkan menghampiri kepada persamaan. Walaubagaimanapun gambar yang dihasilkan tidaklah begitu tepat. Pada keadaan sebenar, warna langit dipengaruhi oleh kesan atmosfera seperti habuk, kabus dan partikel-partikel di udara. Faktor yang kedua adalah berkaitan dengan penukaran warna. Begitu juga perkakasan yang berbeza memerlukan matrik yang berbeza untuk menghasilkan warna.

Nilai maksimum *luminance* yang dikira hanya dengan nilai anggaran. Oleh itu, warna langit yang dihasilkan oleh warna RGB adalah sangat berlainan dengan nilai warna yang sebenar yang dihasilkan oleh warna CIE Yxy. Namun warna CIE Yxy sangat diperlukan kerana model Perez menghasilkan kategori warna CIE Yxy. Selain, warna CIE Yxy ialah warna yang tidak bergantung kepada sesuatu peralatan.

5.3 Pengujian Pengiraan Kedudukan Matahari

Sebagaimana yang diketahui, pengiraan kedudukan matahari bergantung kepada sudut zenith, θ dan sudut azimuth, ϕ dalam radians. Bagi memastikan pengiraan dilakukan dengan tepat, masa bagi matahari terbit dan terbenam dibandingkan dengan data yang sebenar.

Begitu juga, dapat dilihat terdapat perbezaan pada jadual matahari terbit dan terbenam seperti yang ditunjukkan dalam Jadual 5.1 di antara keadaan sebenar dengan simulasi yang dihasilkan. Ini disebabkan oleh pengiraan kedudukan matahari yang tidak begitu tepat.

Daripada hasil pengujian yang di lakukan, maka penentuan warna langit berpandukan kepada kawasan–kawasan yang ingin dikaji tidaklah begitu tepat. Masalah yang dapat di kenalpasti adalah berpunca daripada persamaan untuk mengira kedudukan matahari. Namun imej yang dihasilkn daripada perisian ini, dapat menghasilkan warna langit yang realistik dan boleh digunakan untuk simulasi pada waktu siang.

Rajah 5.15 menunjukkan graf perbandingan data. Daripada graf ini, dapat dilihat bahawa data yang dihasilkan antara kedua–duanya mempunyai corak yang sama. Namun masa yang ditunjukkan oleh graf tidak menunjukkan masa yang sebenar apabila matahari mula terbit dan berakhirnya matahari terbenam.

Walaupun masa yang dihasilkan tidak begitu tepat, perisian tetap dapat memberikan masa yang hampir tepat kepada data sebenar bagi matahari terbit dan terbenam dalam satu tahun.

5.4 Pengujian Penukaran Warna

Sebagaimana yang dibincangkan sebelum ini, nilai warna yang dikira semasa pengiraan warna langit, diberi dalam warna CIE Y_{xy} . Dari sekarang, untuk memaparkannya menggunakan pustaka OpenGL untuk monitor, warna CIE Y_{xy} perlu terlebih dahulu ditukarkan kepada warna RGB. Bahagian ini akan menilai ketepatan hasil output bagi pertukaran warna.

Semasa peringkat pembangunan, perisian penukaran warna dibina untuk membantu dalam pemilihan matrik transformasi. Algoritma bagi menskala nilai *luminance* juga dipertunjukkan. Ia dibangunkan supaya kaedah penukaran yang dibuat adalah tepat. Hasil bagi penukaran warna itu boleh dibandingkan dengan Java applet yang dihasilkan dalam Dawes (1998) dan jadual daripada Adobe-Systems-Inc. (2001).

Warna yang kelihatan adalah warna biru. Tetapi terdapat sedikit perbezaan dalam pencahayaan. Ini menunjukkan warna yang dinyatakan dalam warna RGB adalah anggaran daripada warna sebenar dalam warna CIE. Dari sekarang, hasil dari penukaran warna daripada pengiraan yang dibuat dapat diterima.

5.5 Penutup

Pengujian dibuat untuk menilai pencapaian perisian. Dengan itu kehendak pengguna boleh diperolehi. Perisian yang dihasilkan dipastikan supaya pengiraannya serta hasil outputnya bebas daripada sebarang ralat. Kesan realistik diuji apabila gambar yang diambil daripada kamera eb serta kamera digital dibandingkan. Perbandingan dilakukan antara warna langit yang sebenar dengan warna langit yang dihasilkan. Kesan realistik dapat di paparkan.

BAB 6

KESIMPULAN DAN PERBINCANGAN

6.1 Pendahuluan

Di dalam bab ini, hasil yang telah diperolehi akan diperbincangkan dan satu kesimpulan telah dibuat. Cadangan–cadangan untuk kajian seterusnya dan mempertingkatkan keupayaan sistem.

6.2 Pencapaian Kajian

Secara keseluruhannya, projek ini ingin menghasilkan sebuah warna langit

pada waktu siang yang berpandukan kepada lokasi pemerhati. Beberapa langkah yang diperlukan untuk mendapatkan warna langit iaitu pengiraan kedudukan matahari, pengiraan warna langit dalam kategori warna CIE Yxy dan penukaran warna daripada CIE Yxy kepada warna RGB.

Secara umum, projek ini terdiri daripada beberapa langkah. Langkah pertama adalah menghasilkan model bagi langit iaitu model hemisfera. Ini diikuti dengan pengiraan kedudukan matahari. Kedudukan matahari ditentukan berdasarkan kepada longitud dan latitud pemerhati. GMT juga diperlukan sebagai input. Kedudukan matahari pada permukaan hemisfera ditentukan dengan sudut azimuth dan zenith. Kadang kala pengiraan untuk kedudukan matahari tidak memberikan kedudukan yang tepat. Isu pada masa sekarang ini adalah apabila membuat perbandingan di antara data sebenar matahari terbit dan terbenam dengan data yang dihasilkan oleh perisian ini. Salah satu sebab adalah berkaitan dengan kedudukan matahari pada waktu matahari terbit dan terbenam berbeza pada kawasan yang berbeza. Kedudukan matahari tadi digunakan untuk pengiraan warna langit. Warna yang dihasilkan adalah anggaran sahaja. Ianya telah dikira menggunakan model Perez dengan menghasilkan kategori warna CIE Yxy. Oleh itu, warna yang dikira mungkin berlainan dengan warna yang dipaparkan pada monitor. Monitor yang berlainan menghasilkan warna yang berbeza-beza. Untuk tujuan paparan pada monitor, warna CIE Yxy kemudiannya ditukarkan kepada warna RGB.

Hemisfera yang dihasilkan dengan menggunakan model matematik adalah jauh lebih baik untuk dijadikan model langit. Algoritma yang dihasilkan bukan sahaja dapat menghasilkan segitiga malah dapat menghasilkan segiempat. Projek ini telah memenuhi matlamat dan skop yang dikehendaki. Matlamat yang paling utama adalah untuk membenarkan pengguna melihat keadaan warna langit dengan meletakkan lokasi mereka pada masa yang dikehendaki. Aturcara yang dihasilkan diletakkan di dalam web sebagai applet.

Penghasilan warna langit yang diperolehi menyamai dengan keadaan sebenar pada beberapa keadaan yang berlainan. Ini jelas ditunjukkan dengan perbandingan gambar yang telah dibuat di dalam bab 5. Walaubagaimanapun, keadaan warna langit tidaklah seperti yang ditunjukkan pada setiap masa kerana kita tidak dapat menyangka keadaan langit sebenar pada tarikh dan waktu yang berlainan.

Terdapat beberapa isu yang perlu diperbincangkan untuk mendapatkan warna langit yang sesuai. Penukaran warna langit daripada CIE Yxy kepada RGB tidak begitu tepat. Ianya bergantung kepada monitor yang berlainan. Penukaran warna yang sesuai masih lagi dalam kajian oleh para penyelidik dan keputusan yang benar-benar tepat masih belum ditemui.

Kedudukan matahari memainkan peranan yang penting dalam menentukan warna langit pada masa di lokasi-lokasi yang ingin dilihat. Pengiraan kedudukan matahari yang tidak tepat mengakibatkan waktu matahari terbenam dan waktu matahari naik adalah berlainan daripada keadaan sebenar.

Java 3D merupakan pustaka grafik yang baru diperkenalkan oleh Java yang mana ianya masih lagi baru. Java 3D menggunakan bahasa pengaturcaraan Java sepenuhnya walaupun ianya berdiri di atas pustaka grafik OpenGL atau DirectX. Penggunaan Java 3D merupakan pilihan yang terbaik sekiranya ingin menghasilkan pemvisualan secara 3 dimensi melalui web kerana ianya dapat dijadikan sebagai applet. Pengendaliannya adalah mudah jika dibandingkan dengan VRML yang sering digunakan sebagai pemvisualan yang biasa di gunakan didalam web. Oleh sebab Java 3D adalah sebahagian daripada Java, dan ianya lebih mudah digunakan bersama Java maka penggunaannya dapat membantu pengaturcara-pengaturcara java untuk menghasilkan animasi atau permainan komputer melalui web pada masa sekarang.

6.3 Cadangan

Terlalu banyak perkara perlu diperbaiki dalam projek ini. Perisian ini boleh ditambah lagi dengan memodelkan awan, pergunungan dan laut selain daripada langit.

Kajian ini hanya tertumpu kepada penghasilan warna langit yang tidak berawan. Untuk mendapatkan keadaan langit pada setiap keadaan, maka kajian terperinci perlu dilaksanakan. Hasil daripada kajian ini dapat menunjukkan keadaan langit untuk menentukan keadaan cuaca secara 3 dimensi. Dalam pada itu, ianya juga sesuai untuk dimuatkan di dalam permainan simulasi kapal terbang di mana gambaran keadaan langit dapat ditunjukkan sekiranya kapal terbang terbang pada altitud yang tinggi.

Sekiranya dapat menghasilkan awan bersama langit keadaan yang lebih realistik akan dapat dihasilkan. Namun perlu diambil kira bahawa warna langit pada keadaan yang tidak berawan dengan keadaan yang berawan adalah berbeza. Justeru itu, untuk mendapatkan keadaan yang lebih realistik, pengiraan warna yang mengambil kira kesan serakan warna ke atas awan perlu.

Untuk memperbaiki pada masa akan datang, supaya perisian ini bukan sahaja memodelkan warna langit malah unsur-unsur yang lain juga turut dimodelkan, terdapat dua bahagian yang perlu diperbaiki. Pertama, tambahan boleh dilakukan ke atas perisian. Kedua, mengemukakan teknik yang lebih baik atau memperbaiki teknik ini.

Pensteksturan awan ke atas langit boleh dibuat dengan menggunakan fungsi Perlin. Fungsi Perlin dapat menghasilkan awan secara rawak. Ianya dapat menghasilkan awan yang bergerak berbanding dengan awan yang statik. Untuk maklumat lanjut, sila rujuk Perlin (2001)[14] dan Elias (2001)[4]. Dengan menggunakan pensteksturan imej hanya dapat menghasilkan kesan 2D. Untuk mendapatkan kesan yang lebih realistik, permodelan partikel (Reeves (1983))[15] atau metaball (dobashi et al. (1999))[3] boleh digunakan.

Untuk mendapatkan suasana yang lebih nyata, permukaan muka bumi dan bukit bukau boleh diletakkan bersama. Interaksi antara cahaya matahari dengan permukaan bumi dapat menghidupkan lagi dibunakan. Oleh itu, ianya dapat menghasilkan warna yang berbeza pada waktu yang berlainan

Dalam projek ini, model bagi matahari tidak disertakan. Matahari boleh dimodelkan dengan meletak tekstur pada kedudukan matahari. Salah satu cara adalah dengan memodelkan matahari mengikut kedudukan matahari, saiz yang bersesuaian dan bergerak berdasarkan kedudukan matahari. Dalam pada itu, suasana kabus yang menyerupai keadaan sebenar dapat diberikan kesan yang realistik.

Hasil daripada kajian ke atas warna langit ini dapat dijadikan rujukan dan hasilnya dapat ditunjukkan kepada para penyelidik dengan lebih mudah kerana ianya dapat dimuatkan di dalam web. Justeru itu, adalah lebih baik sekiranya penggunaan Java diperluaskan kerana keupayaan Java yang dapat menghasilkan objek 3 dimensi dengan penggunaan Java 3D dan berkeupayaan untuk merentasi platform yang berlainan.

6.4 Kesimpulan

Warna langit yang dihasilkan dapat memaparkan keadaan langit yang lebih realistik. Ianya hanya dapat menghasilkan keadaan yang hampir sama dengan keadaan sebenar apabila keadaan langit tidak berawan. Sekiranya applikasi yang ingin dibangunkan memerlukan keadaan langit yang realistik, hasil daripada projek ini dapat digunakan.

Penggunaan web sebagai media yang terpenting pada masa kini telah membuatkan manusia berusaha untuk menambahkan lagi keupayaan web selain daripada memaparkan kandungan html yang biasa. Penggunaan teknologi 3D di dalam web dapat membantu para penyelidik memaparkan hasil kajian mereka melalui web supaya ianya dapat dijadikan sebagai sumber rujukan.

Bibliografi

Adobe-Systems-Inc.. Technical guides : Color management in photoshop 5.x- cmyk setup. <http://www.adobe.com/support/techguides/color/photoshop/cmyk-set.html>, 2001. cited 10th August 2001.

Apple. (1996). *Advanced color imaging on the macos*. Apple Computer Inc.

BBC-Online. England webcam – bridlington. <http://www.bbc.co.uk/england/webcams/regional/bridlington/>, 2001. cited 4th Sept 2001.

Berlin, B.; Kay, P. (1969). *Basic color terms*. Berkeley. University of California Press. Reprinted 1991.

Blinn, J.F. (1982). *Light reflection functions for simulation of clouds and dusty surfaces*. 16, 21-29.

CIE. International commission on illumination. <http://www.cie.co.at/cie/home.html>, 2001. cited 1st August 2001.

Dawes, B. Color metric converter. <http://www.colorpro.com/info/tools/convert.htm>, 1998. cited 2nd August 2001.

Dobashi, Y.; Nishitra, T.; Yamashita, H.; Okita, T. (1999). Using meatballs to modeling and animate clouds from satellite images. *The Visual Computer*. 15 : 471-482.

Dobashi, Y.; Nishita, T.; Kaneda, K.; Yamashita, H. (1997). A fast display method of sky colour using basis functions. *The Journal of Visualization and Computer Animation*. 8(3) : 115-127.

Elias, H. Perlin noise. http://freespace.virgin.net/hugo.elias/models/m_perlin.htm, 2001. cited 1st August 2001.

France-telecom. Webcams paris. <http://webcm.mkm-wpe.net/>, 2001. cited 4th Sept 2001.

Glassner, A. (1989). *An introduction to ray tracing*. Academic Press. 329 pp.

Hill, F. (2001). *Computer graphics using opengl*. 2nd ed. Prentice Hall, 922 pp.

Hollingworth, J.; Butterfield, D.; Swart, B.; Allsop, J. (2001). *C++ builder 5 developer's guide*. Sams Publishing, 1394 pp.

- Iqbal, M. (1983). *An introduction to solar radiation*. Academic Press, 390 pp.
- Jansen, H.W.; F.; Dorsey, J.; Stark, M.M.; Shirley, P.; Promoze, S. (2001). A physically-based night sky model. *Computer Graphics (SIGGRAPH '2001 Proceedings)*.
- Kaneda, K.; Okamoto, T.; Nakamae, E.; Nishita, T. (1991). Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. *The Visual Computer*. 7(5) : 247-258.
- Klassen, R.V. (1987). Modelling the effect of the atmosphere on light. *ACM Transactions on Graphics*. 6(3) : 215-237.
- Lake, A.; Marshall, C.; Harris, M.; Blackstein, M. Stylised rendering techniques for scalable real-time 3d animation. <ftp://download.intel.com/ial/3dssoftware/toon.pdf>, 2001. cited 9th July 2001.
- Max, N.L. (1986). Atmospheric illumination and shadows. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20 : 117-124.
- Naval, U. Us naval observatory – world map of time zone. <http://aa.usno.navy.mil/>, 2001. cited 31st August 2001.
- Nishita, T.; Nakamae, E.; Dobashi, Y. (1996). Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In Rushmeier, H.; ed., SIGGRAPH '96 Conference Proceeding, Annual Conference Series. 04-09 August 1996, Held in New Orleans, Louisiana : Addison Wesley. 379-386.
- Nishita, T.; Nakamae, E.; Dobashi, Y. Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In Rushmeier, H.; ed., SIGGRAPH '96 Conference Proceeding, Annual Conference Series.; 04-09 August : Held in New Orleans, Louisiana : Addison Wesley, 1996. 379-386.
- Perez, R.; Seals, R.; Michalsky, J. (1993). All-weather model for sky luminance distribution – preliminary configuration and validation. *Solar Energy*. 50(3) : 235-245.
- Perlin, K. Perlin noise. <http://mrl.nyu.edu/perlin/>, 2001. cited 1st August 2001.
- Poynton, C. Frequently asked question about color. <http://www.inforamp.net/poynton>, 1999 cited 5th August 2001.
- Preetham, A.J.; Shirley, P.; Smith, B. (1999). A practical analytic model for daylight. *Computer Graphics (SIGGRAPH '99 Proceedings)*. 91-100.
- Reeves, W.T. (1983). Particles systems – a technique for modeling a class fuzzy objects. *Computer Graphics (SIGGRAPH '1983 Proceedings)*. 17 : 359-376.

- Shaw, J.A. (1996). The light touch. *Optics and Photonics News*. 8(4) : 54-55.
- Van de Hulst, H.C. (1957). *Light scattering by small particle*. Chapman and Hall, 470 pp.
- Walter, B. (2000). Simulation and visualization of atmospheric light phenomena induced by light scattering. International Conference of Environmental Modeling and Simulation.
- Watt, A. (2000). *3d computer graphics*. 3rd ed. Addison-Wesley Longman Inc, 570 pp.
- Watt, A. (2001). *3d games : Real-time rendering and software technology*. Addison Wesley Longman Inc, 800 pp.
- Weather-Channel. Daily climate statistics.
<http://www.weather.co.uk/weather/climatology/UKXX0075.htm>, 2001. cited 31st August 2001
- Woste, L. Mie scattering. <http://www.physik.fu-berlin.de/ag-woestemie.html>, 2001. cited 21st August 2001.
- Wright, R.S.; Sweet, M. (1999). *OpenGL super bible*. 2nd ed. Waite Group Press, 696 pp.

Manual Pengguna

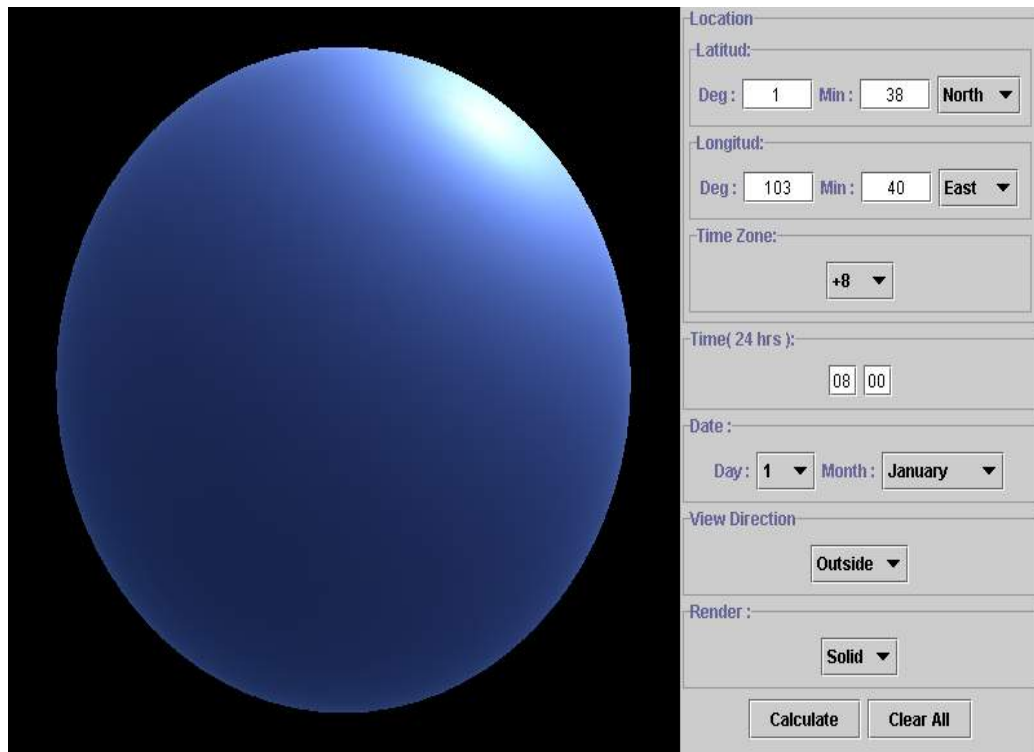
A.1 Pemasangan (Instalasi)

Secara umumnya aturcara ini dihasilkan di dalam bentuk applet. Ianya dapat dimuatkan bersama dengan html. Untuk menggunakan aturcara ini, pengguna hanya perlu pergi ke website dan kelas – kelas untuk applet akan di pindah turunkan. Pelayar yang digunakan mestilah mempunyai kebolehan untuk melarikan java. Beberapa program perlu di pindah turunkan dari internet terlebih dahulu untuk melihat simulasi yang dihasilkan. Pengguna perlu memindah turun Java Plug- In dan Java 3D Runtime (RT). Terdapat 2 versi Java 3D Runtime iaitu Java 3D RT (OpenGL) dan Java 3D (Direct X). Pengguna boleh memilih mana – mana versi Java 3D yang dikehendaki tetapi disyorkan supaya memilih Java 3D RT (OpenGL) kerana OpenGL telah ditampung oleh kebanyakan platform.

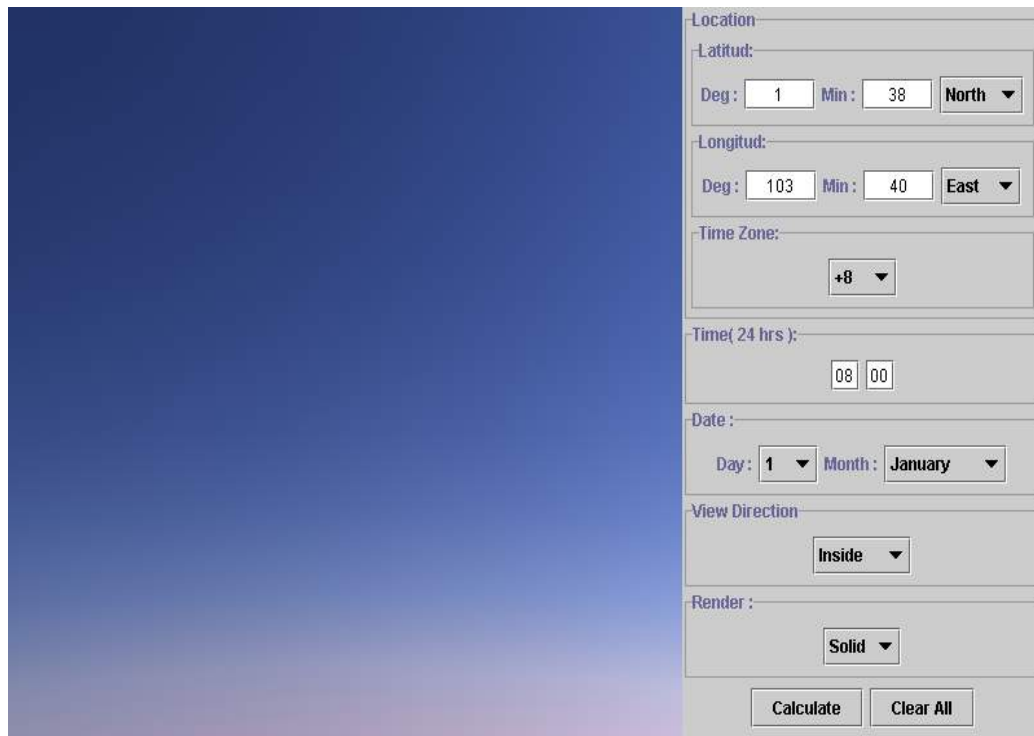
Tujuan utama memindah turun Java Plug –In adalah untuk membolehkan aturcara yang dijadikan sebagai applet tadi menggunakan Java Virtual Machine (JVM) yang baru . Pengguna perlu memindah turun Java Plug – In v 1.3.0 ke atas dengan menggunakan Java 2 (TM) Runtime Enviroment v 1.3.0 keatas. Setelah kesemua program tadi dipindah turun, aturcara simulasi dapat dilihat.

Untuk mengetahui keadaan warna langit yang bagi sesuatu lokasi. Pengguna perlu meletakkan kedudukan longitud dan latitud sesuatu kawasan. Pengguna juga perlu meletakkan masa dan tarikh serta perlu mengetahui zon masa sesuatu kawasan. Tekan butang ‘calculate’ untuk memulakan pengiraan kemudian simulasi dimulakan untuk mendapatkan warna langit.

A.2 Penggunaan Aturcara



Secara 'default' pengguna hanya dapat melihat pandangan daripada bahagian luar hemisfera. Untuk mendapatkan pandangan daripada bahagian dalam hemisfera, ubah menu view direction kepada bahagian 'inside'. Warna langit yang ditunjukkan disini berada pada lokasi di Johor Bahru iaitu $1^{\circ} 38''$ Utara, $103^{\circ} 40''$ Timur.



The image shows a software interface with a blue gradient background on the left and a grey control panel on the right. The control panel is organized into several sections:

- Location:**
 - Latitud:** Deg: Min: **North** ▼
 - Longitud:** Deg: Min: **East** ▼
 - Time Zone:** ▼
- Time(24 hrs):**
- Date:** Day: ▼ Month: ▼
- View Direction:** ▼
- Render:** ▼

At the bottom of the control panel are two buttons: **Calculate** and **Clear All**.

Untuk melihat pandangan dari dalam, pengguna hanya perlu memilih menu 'view direction' kepada 'inside'. Pengguna boleh melihat sekeliling 360 darjah dengan menggunakan butang anak panah pada papan kekunci dan pastikan klik pada bahagian sebelah kanan terlebih dahulu.

Untuk menentukan pada koordinat yang lain pula, tekan butang 'Clear All' dan semua nilai input text akan dipadam. Pengguna perlu mengetahui kedudukan lokasi terlebih dahulu.