

Aplikasi Graf pada Variasi Interpolasi *Keyframing* Animasi untuk Meningkatkan Kualitas Citra Bergerak

Edward Alexander Jaya - 13517115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517115@std.stei.itb.ac.id

Abstrak—Teori Graf merupakan cabang ilmu Matematika Diskrit yang banyak sekali digunakan dalam kehidupan sehari-hari. *Keyframing* adalah salah satu teknik paling sederhana untuk membuat animasi. Adapun *keyframing* menggunakan interpolasi dan pemodelan teknik ini dapat divisualisasikan dengan graf berarah. Namun, *keyframing* memiliki beberapa kelemahan jika interpolasi yang digunakan bukanlah interpolasi tingkat tinggi, karena akan membuat gerakan yang kurang realistis. Variasi dari interpolasi cukup banyak, namun pada makalah ini akan dibahas dua variasi interpolasi *keyframing* yang paling banyak digunakan, yaitu interpolasi lanjar dan interpolasi *Spline*.

Kata kunci—animasi, graf, interpolasi, keyframe

I. PENDAHULUAN

Teori Graf adalah cabang ilmu Matematika Diskrit yang memiliki banyak kegunaan dalam kehidupan sehari-hari, salah satunya adalah aplikasinya dalam video. Graf dinyatakan dalam bentuk titik yang disebut juga dengan simpul. Titik tersebut merepresentasikan objek. Jika objek tersebut memiliki hubungan dengan objek-objek lain, maka objek yang dinyatakan dengan titik tersebut memiliki garis dengan titik-titik lain. Garis tersebut disebut juga dengan sisi.

Video didefinisikan sebagai citra / gambar yang bergerak. Gambar bergerak tersebut merupakan hasil dari kombinasi keterurutan gambar-gambar diam. Setiap gambar diam memiliki informasi berupa visualisasi objek, misalnya objek berupa manusia dan meja. Objek yang divisualisasikan tersebut memiliki posisi relatif terhadap dimensi gambar. Biasanya posisi relatif objek pada gambar yang satu dengan gambar yang lain berbeda, sehingga ketika gambar-gambar tersebut digabungkan secara terurut lalu dilihat pada kecepatan tertentu, tercipta gambar yang bergerak.

Gambar-gambar pada video tersebut bisa merupakan foto-foto, artinya gambar-gambar tersebut nyata. Gambar selain hasil foto dapat juga dihasilkan melalui perangkat lunak pada komputer, seperti *Adobe Photoshop* dan *Adobe Illustrator*. Jika gambar dihasilkan dengan komputer, maka video tersebut lebih dikenal dengan sebutan animasi atau *Computer Generated Imagery (CGI)*.

Animasi pada zaman sekarang sangat sering dipakai dalam pembuatan film-film, terutama film-film Hollywood yang menggunakan *CGI* yang terlihat realistis dan digabungkan dengan video nyata. Kegunaan animasi ini adalah

memvisualisasikan imajinasi sutradara agar dapat dimunculkan di film secara nyata.

Salah satu aplikasi graf pada animasi digunakan untuk menyatakan keadaan visual objek. *Keyframing* adalah teknik untuk mendefinisikan keadaan objek awal dan akhir yang kemudian membentuk transisi sehingga terbentuk gerakan dari keadaan awal hingga akhir.

Karena *keyframing* ini membuat transisi dari keadaan awal objek sampai keadaan akhir objek, terdapat banyak sekali kemungkinan gerakan yang bisa dihasilkan dari keadaan awal ke akhir tersebut. Misalnya saja, posisi awal objek terdapat di ujung kiri gambar dan posisi akhir objek terdapat di ujung kanan gambar, Objek tersebut dapat bergerak ke atas dahulu baru ke kanan gambar. Atau, objek tersebut dapat bergerak ke bawah lalu ke kanan gambar, Atau, objek tersebut dapat berotasi lalu baru bergerak ke kanan gambar.

Karena transisi yang dihasilkan banyak sekali, perlu dibuat relatif banyak simpul baru di antara posisi awal dan posisi akhir, agar gerakan gambar bisa dibuat secara tepat. Interpolasi yang didefinisikan sebagai teknik memasukkan titik atau nilai baru berdasarkan nilai yang sudah diketahui berperan pada proses ini. Dengan bantuan intelegensia buatan, interpolasi oleh komputer secara otomatis tentunya akan sangat membantu dalam pembuatan simpul baru di antara posisi awal dan akhir.

Pada makalah ini, penulis akan membahas lebih lanjut mengenai aplikasi graf pada interpolasi *keyframing* pada animasi sehingga gerakan yang dihasilkan sesuai dengan keinginan pembuat animasi, agar dapat tercipta kualitas citra yang lebih baik.

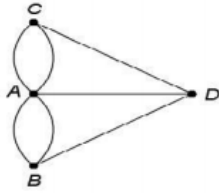
II. TEORI DASAR

A. Definisi Graf

Graf dinyatakan sebagai pasangan himpunan (V, E) . Secara definisi:

- V adalah himpunan tidak kosong dari simpul-simpul.
- E adalah himpunan sisi-sisi.

Atau jika dibuat notasi singkatnya, $G = (V, E)$. Simpul dapat dinyatakan dengan bentuk titik, yang merepresentasikan objek. Sedangkan hubungan objek ke objek-objek lain dinyatakan dengan sisi. Sisi divisualisasikan dengan bentuk garis. Contoh dari graf adalah sebagai berikut.



Gambar I: Ilustrasi graf^[1]

Simpul pada graf dapat diberi nama dengan huruf seperti a,b,c,... atau dengan bilangan asli 1,2,3,..., atau kombinasi dari keduanya. Pada contoh gambar di atas, simpul ditunjukkan dengan huruf A, B, C,D. Sedangkan sisi e yang menghubungkan sebuah simpul v_a dengan v_b dapat ditulis dengan

$$e = (v_a, v_b)$$

Pada gambar di atas, salah satu sisi ditunjukkan dengan

$$e = (A,C).$$

B. Jenis-Jenis Graf

Jenis graf dapat dibagi berdasarkan beberapa kategori. Kategori pertama, graf dapat dikelompokkan berdasarkan ada atau tidaknya sisi ganda atau sisi kalang (*loop*). Kategori kedua, graf dapat dikelompokkan berdasarkan jumlah simpulnya. Kategori ketiga, graf dapat dikelompokkan berdasarkan arah pada sisi. Sebutan-sebutan pada jenis graf di bawah ini akan digunakan lebih banyak di upabagian berikutnya.

Berdasarkan ada atau tidaknya sisi ganda atau sisi kalang, graf dapat dibagi menjadi:

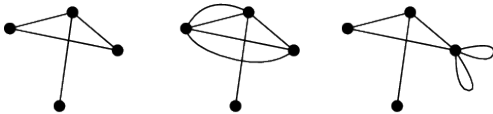
1. Graf Sederhana

Graf ini tidak memiliki kalang maupun sisi ganda.

2. Graf Tak Sederhana

Graf ini memiliki kalang ataupun sisi ganda.

Contoh dari graf pada kategori ini adalah:



Gambar II: Graf Sederhana dan Graf Tak Sederhana^[2]

Pada gambar di atas, graf paling kiri adalah graf sederhana. Graf yang berada di tengah mengandung sisi ganda dan graf paling kanan mengandung kalang (*loop*). Artinya, graf tengah dan graf paling kanan adalah graf tidak sederhana.

Berdasarkan jumlah simpulnya, graf dapat dibagi menjadi:

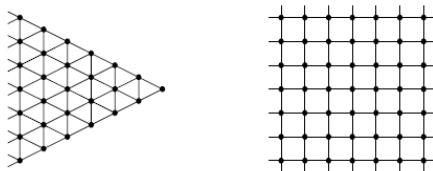
1. Graf Berhingga

Graf ini memiliki jumlah simpul yang berhingga.

2. Graf Tak Berhingga

Graf ini memiliki jumlah simpul yang tak berhingga.

Contoh dari graf pada kategori ini adalah:



Gambar III: Graf Tak Berhingga^[1]

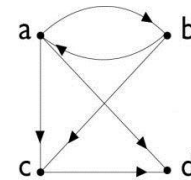
Berdasarkan ada atau tidaknya arah pada sisi, graf dapat dikelompokkan menjadi:

1. Graf Tak Berarah

Graf ini adalah graf yang sisinya memiliki arah.

2. Graf Berarah

Graf ini memiliki sisi yang memiliki arah. Arah direpresentasikan dengan panah. Pada graf ini, terdapat istilah *simpul asal* dan *simpul terminal*. Simpul asal memiliki sisi berarah menuju ke simpul terminal. Adapun sisi seringkali disebut busur pada graf berarah. Istilah ini lebih jelas bila direpresentasikan dengan gambar berikut:



Gambar IV: Graf Berarah^[3]

Salah satu simpul asal di atas adalah simpul a . Karena a memiliki sisi berarah ke b , b disebut dengan simpul terminal. Pada pembahasan mengenai keyframing animasi, graf berarah akan lebih banyak digunakan karena setiap posisi yang direpresentasikan dengan simpul berkorespondensi dengan waktu.

C. Terminologi Dasar Graf

1. Bertetangga

Istilah ini digunakan pada dua simpul yang terhubung dengan sebuah sisi.

2. Bersisian

Istilah ini digunakan pada sisi e yang menghubungkan v_a dan v_b . e disebut juga bersisian dengan v_a dan v_b .

3. Derajat

Derajat pada suatu simpul di graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Pada graf berarah, terdapat derajat masuk v , yaitu jumlah busur yang masuk ke simpul v . Selain itu, terdapat derajat keluar v , yaitu jumlah busur yang keluar dari simpul v . Derajat simpul v adalah derajat masuk v + derajat keluar v .

4. Lintasan

Lintasan yang memiliki panjang n dari simpul v_0 ke v_n adalah barisan selang-seling simpul-simpul dan sisi-sisi yang memiliki sekuens $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$. Sehingga terbentuk $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$.

5. Sirkuit

Sirkuit adalah lintasan yang dimulai dari simpul n dan diakhiri pada simpul n lagi.

6. Terhubung

Pada graf tak berarah, simpul v_1, v_2 dikatakan terhubung bila terdapat lintasan dari v_1 ke v_2 . Adapun graf terhubung adalah graf dengan setiap pasang simpul terhubung.

Pada graf berarah G , jika graf tak berarahnya terhubung, G adalah graf terhubung. Jika terdapat lintasan berarah dari v_1 ke

v_j dan v_k ke v_i , kedua simpul tersebut terhubung kuat. Jika tidak, kedua simpul tersebut terhubung lemah.

7. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton adalah lintasan yang melalui setiap simpul di dalam graf tepat satu kali. Sirkuit Hamilton adalah lintasan Hamilton yang kembali ke simpul asal.

8. Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).

D. Interpolasi

Interpolasi adalah komputasi dari beberapa titik atau nilai di antara titik atau nilai yang sudah diketahui. Titik atau nilai tersebut dibangkitkan berdasarkan titik atau nilai lain yang berdekatan.

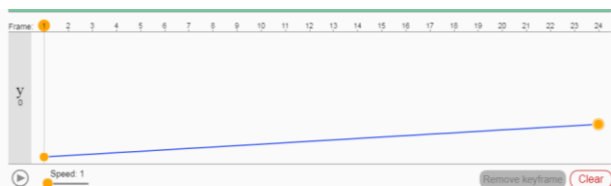
Atau dalam bahasa lain, jika diberikan fungsi $f = f(x)$ dengan beberapa nilai yang sudah diketahui. Jika $S = \{x_0, x_1, \dots, x_n\}$, interpolasi membangkitkan nilai $f(x)$ dengan nilai x yang tidak berada pada himpunan S menggunakan nilai x yang sudah ada pada himpunan S .

Interpolasi dapat dibagi menjadi banyak jenis, namun pada makalah ini hanya akan dibahas mengenai *interpolasi linjar* serta *interpolasi spline*.

1. Interpolasi Linjar

Interpolasi paling sederhana karena hanya mengambil data dari titik atau nilai yang paling dekat dengan titik atau nilai yang diproduksi. Interpolasi jenis ini tidak berbentuk polinomial. Jika digambar pada graf posisi terhadap waktu, hanya akan terbentuk garis lurus / linjar.

Kelebihan interpolasi jenis ini adalah hemat memori, namun kekurangannya, animasi yang dihasilkan tidak mulus.



Gambar V: Interpolasi Linjar [5]

2. Interpolasi Spline

Spline didefinisikan sebagai salah satu tipe kurva mulus dalam ruang 2D / 3D. Artinya, interpolasi yang dilakukan bukan dengan hanya fungsi linjar saja, tetapi dapat dilakukan dengan fungsi polinomial.

Contoh fungsi polinomial tersebut adalah sebagai berikut:

Polinom Posisi :

$$p(t) = at^3 + bt^2 + ct + d$$

Polinom Kecepatan :

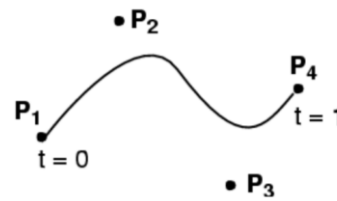
$$v(t) = 3at^2 + 2bt + c$$

Polinom Akselerasi:

$$a(t) = 6at + 2b$$

Akibatnya, animasi yang dihasilkan tentunya akan lebih mulus. Salah satu hasil dari interpolasi ini adalah *Bezier Curve*. *Bezier Curve* adalah kurva yang dihasilkan pada interpolasi ini. Dalam grafika komputer, salah satu jenis dari *Bezier Curve* adalah *Cubical Bezier Curve* yang menggunakan empat titik (disebut

juga sebagai titik kontrol). Contoh dari *Bezier Curve* adalah sebagai berikut:



Gambar VI: *Cubical Bezier Curve* [6]

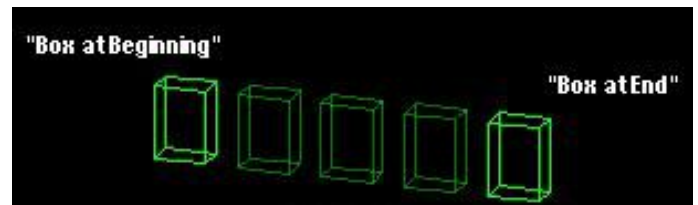
E. Keyframing

Keyframing adalah suatu teknik untuk menganimasikan objek berdasarkan bingkai / *frame* kondisi awal dan kondisi akhir. "Key" pada *Keyframing* menyatakan bingkai kunci yang menjadi landasan animasi dilakukan. Contohnya adalah sebagai berikut:



Gambar VII: (a) Bingkai Pertama / First Frame, (b) Bingkai Terakhir / Last Frame [7]

Pada gambar di atas, sebuah objek berupa boks awalnya berada di kiri (a). Kondisi terakhir boks akhirnya berada di kanan (b). *Keyframing* membuat frame baru secara otomatis berdasarkan kondisi pada gambar (a) dan (b). Dapat disimpulkan bahwa *keyframe* menggunakan proses interpolasi untuk menghasilkan bingkai baru berdasarkan kedua bingkai kunci tersebut.



Gambar VIII: Bingkai yang dihasilkan oleh *keyframing* [7]

Pada gambar VIII, terdapat tiga bingkai baru yang diproduksi oleh *keyframing*, menjadikan terciptanya lima total bingkai. Jika misalnya pengguna ingin boks dari Gambar VII(a) bergerak ke atas sebanyak 100 pixel terlebih dahulu lalu baru bergerak ke kondisi akhir (Gambar VII(b)), maka tambahkan bingkai baru berupa boks yang berada pada posisi 100 pixel di atas kondisi Gambar VII(a). Maka tercipta animasi yang berbeda.

III. INTERPOLASI KEYFRAMING

A. Permasalahan *Keyframing*

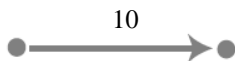
Di dalam dunia animasi, *keyframing* dengan menggunakan dua bingkai kunci yang masing-masingnya menyatakan keadaan

awal dan keadaan akhir tidaklah cukup. Perubahan visual dari keadaan awal sampai keadaan akhir disebut juga dengan transisi. Seperti yang telah disinggung pada bagian pendahuluan, terdapat banyak sekali transisi yang dapat dihasilkan dari keadaan awal ke keadaan akhir. Kecepatan yang dihasilkan oleh keyframe tanpa interpolasi tingkat tinggi juga konstan sehingga gerakan yang dihasilkan tidak bagus. Sebagai kesimpulan, gerakan yang dihasilkan dengan teknik *keyframing* dengan bingkai kunci yang terlalu sedikit tidak terlalu realistis.

Namun, telah ditemukan salah satu solusi untuk membuat *keyframe* menjadi semakin realistis, yaitu dengan teknik interpolasi yang akan dibahas pada upabagian selanjutnya.

B. Pemodelan Keyframing Lanjar Dengan Graf

Setiap simpul pada graf merepresentasikan keadaan di suatu bingkai (*frame*) dan setiap sisinya merepresentasikan transisi atau perubahan visual. Karena *keyframing* memiliki faktor waktu sebagai landasan terjadinya transisi, maka graf yang dibuat merupakan graf berarah dan berbobot. Bobot pada setiap sisi menyatakan waktu, biasanya dalam satuan *frame*. Selain itu, graf memiliki lintasan Hamilton, karena setiap simpul pasti bisa dilalui tepat satu kali. Sebenarnya satu simpul bisa memiliki derajat lebih dari satu, namun untuk menyederhanakan, setiap simpul hanya memiliki derajat satu. Penyederhanaan ini digunakan untuk merepresentasikan *keyframing* pada aplikasi sebenarnya, yaitu transisi dari setiap pasangan *keyframe* hanya bisa dimainkan satu kali pada satu waktu. Graf minimal memiliki simpul sebanyak dua, sebagai syarat agar animasi terjadi. Penggambaran visual pada graf diperlihatkan di bawah ini:



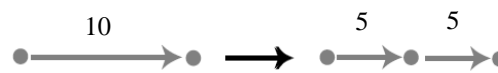
Gambar IX: Graf Berarah dengan Bobot 10 *Frame* ^[8]

Pada graf ini, jika graf memiliki n bilangan bulat positif simpul yang merepresentasikan bingkai kunci, terdapat $n-1$ sisi berarah yang merepresentasikan transisi. Hal ini dapat dibuktikan dengan menggunakan induksi matematika. Basis dari induksi ini yaitu graf dengan dua simpul. Karena graf memiliki dua simpul, graf pasti memiliki satu sisi. Sekarang perlu ditunjukkan langkah induksinya. Andaikan untuk setiap n simpul terdapat $n-1$ sisi berarah, maka perlu ditunjukkan bahwa $n+1$ simpul memiliki n sisi berarah. Untuk membuktikan ini, dapat ditunjukkan bahwa $n+1$ simpul memiliki $(n+1)-1 = n$ sisi berarah. Karena basis dan langkah induksi benar, untuk semua bilangan bulat positif $n > 1$ simpul, terdapat $n-1$ sisi berarah.

C. Pemodelan Interpolasi Keyframing Lanjar dengan Graf

Pada kasus ini, jika pada awalnya graf memiliki m buah simpul dan akan dimasukkan total n buah *frame* baru di antara pasangan simpul bersisian yang sembarang, maka total waktu untuk menjalankan animasi tersebut adalah tetap. Hal ini bisa terjadi karena interpolasi adalah proses memasukkan titik atau nilai (pada kasus ini simpul) di antara pasangan simpul yang sudah diketahui. Namun, waktu untuk menjalankan animasi di

antara simpul lama dan simpul baru yang bersisian tentunya lebih sedikit daripada waktu untuk menjalankan animasi di antara pasangan simpul lama (sebelum dimasukkan). Hal ini dapat lebih jelas ditunjukkan dengan ilustrasi sebagai berikut:



Gambar X: Interpolasi Graf ^[8]

Pada gambar di atas, gambar sebelah kiri memiliki bobot dua simpul dengan bobot sepuluh *frame*. Setelah dimasukkan satu simpul baru di antara dua simpul yang lama, total bobot adalah sepuluh juga, tetapi jarak dari simpul paling kiri ke simpul tengah adalah lima, lebih sedikit dari sepuluh.

Atau dalam notasi matematika secara umum, jika n simpul baru dimasukkan ke dalam dua simpul yang bersisian pada awalnya, maka terdapat $n + 2 - 1 = n + 1$ sisi berdasarkan induksi matematika yang sudah dijelaskan sebelumnya.

Jika dua simpul awal yang bersisian tersebut memiliki bobot x_{tot} dan setiap simpul i beserta pasangannya simpul $i + 1$ (bersisian dengan simpul i) memiliki bobot x_i dengan i adalah bilangan bulat dan $0 < i < n + 2$, maka :

$$x_1 + x_2 + \dots + x_{n+1} = x_{tot} \quad (1)$$

Bobot pada setiap x_i di antara range $0 < i < n + 2$ tidak harus sama, tetapi bisa berbeda asalkan memenuhi persamaan (1).

D. Perhitungan pada Interpolasi Lanjar

Pada beberapa gambar sebelumnya, graf memiliki sisi yang bentuknya bukan kurva, melainkan lanjar. Interpolasi lanjar adalah variasi interpolasi dengan pemodelan graf ini. Pada graf dengan bentuk sisi lanjar, animasi yang dihasilkan memiliki kecepatan konstan atau tidak memiliki variasi kecepatan.

Informasi keadaan pada setiap simpul bisa bermacam-macam, mulai dari keadaan posisi, lalu intensitas cahaya, rotasi, dan parameter-parameter lainnya. Pada bagian ini, akan dijelaskan perhitungan sederhana mengenai interpolasi lanjar.

Pada graf dengan n simpul (total bobot x_{tot}) yang sudah diinterpolasi, terdapat $n-1$ sisi, sesuai dengan pembuktian induksi matematika. Untuk menghitung seberapa besar perubahan keadaan yang ada di setiap transisi pasangan simpul i dan $i + 1$, dengan range $0 < i < n$, carilah keadaan di simpul i dan $i+1$ tersebut. Jika pasangan tersebut memiliki bobot x_i , maka persamaan mencari kecepatan perubahan keadaan :

$$\text{perubahan}_i = \frac{\text{keadaan}_{i+1} - \text{keadaan}_i}{x_i} \quad (2)$$

Sedangkan untuk mencari perubahan rata-rata pada graf dengan n simpul:

$$\frac{\text{perubahan}_1 + \text{perubahan}_2 + \dots + \text{perubahan}_n}{n-1} \quad (3)$$

Sebagai contoh, setiap simpul pada graf menyimpan informasi berupa keadaan posisi x dan y . Diketahui graf

memiliki tiga simpul. Pada simpul pertama, cari posisi x , misalnya $x = 100$ pixel dan cari posisi y , misalnya $y = 200$ pixel. Lalu pada simpul kedua, didapat bahwa $x = 150$ pixel dan $y = 250$ pixel. Diketahui juga bobot di antara dua simpul tersebut adalah sepuluh *frame*. Maka, kecepatan perubahan posisi x di antara simpul pertama dan kedua adalah :

$$\frac{(150-100) \text{ pixel}}{10 \text{ frame}} = 5 \frac{\text{pixel}}{\text{frame}} \quad (4)$$

Sedangkan kecepatan perubahan posisi y adalah:

$$\frac{(250-200) \text{ pixel}}{10 \text{ frame}} = 5 \frac{\text{pixel}}{\text{frame}} \quad (5)$$

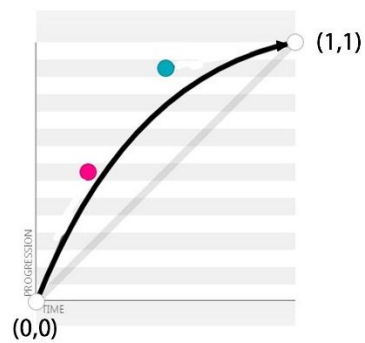
Diketahui juga pada simpul ketiga posisi $x = 250$ pixel dan posisi $y = 400$ pixel, serta bobot antara simpul kedua dan ketiga adalah 10 *frame*. Maka dengan persamaan (2) didapat perubahan posisi $x = 10$ pixel/frame dan perubahan posisi $y = 15$ pixel/frame. Menggunakan persamaan (3), didapat rata-rata perubahan posisi x adalah $(5+10)/2 = 7.5$ pixel/frame dan perubahan posisi y adalah $(5+15)/2 = 10$ pixel/frame.

Film untuk ditayangkan di bioskop biasanya memiliki *frame rate* sebesar 24 *frame per second* (*fps*). Pada contoh di atas, animasi memiliki total bobot 20 *frame*. Maka, animasi yang dijalankan memiliki waktu $20/24 = 0,83$ detik saja!

Persamaan (2) berguna untuk menentukan seberapa cepat animasi akan bergerak di antara dua *frame* dan persamaan (3) berguna untuk mengetahui kecepatan objek berubah sepanjang animasi. Jika perubahan rata-rata per *frame* besar, animasi bergerak cepat dan kadang gerakan ini tidak terlihat realistis, sehingga seringkali ditambahkan *blur* pada gerakan tersebut. Aplikasi dari kedua persamaan ini terdapat dalam perangkat lunak pembuat animasi, seperti Adobe After Effects dan Adobe Flash.

E. Pemodelan dan Perhitungan pada Interpolasi Keyframing Spline

Interpolasi linier membuat animasi yang kurang realistis akibat tidak adanya variasi kecepatan. Interpolasi *Spline* yang menghasilkan *Bezier Curve* dapat memuat variasi kecepatan tersebut. *Bezier Curve* dapat dibagi menjadi beberapa jenis, yaitu *Quadratic Bezier Curve* dan *Cubical Bezier Curve*. Pada bagian ini akan dibahas mengenai *Cubical Bezier Curve*. Pemodelan *Bezier Curve* ini mirip seperti gambar IX, namun perlu diketahui bahwa bentuk sisi bukanlah linier tetapi kurva kubik (seperti gambar VI). Maka model yang dihasilkan adalah graf dengan minimal empat simpul. Dua simpul menyatakan masing-masing keadaan awal dan keadaan akhir pada graf, dan dihubungkan oleh satu sisi berarah sebagai representasi transisi. Dua simpul lainnya menyatakan *control points* yang tidak akan dilewati oleh sisi tersebut, namun berefek pada bentuk sisi di antara keadaan awal dan akhir. Pada contoh berikut akan diperlihatkan *Bezier Curve* dengan keadaan setiap simpul adalah keadaan waktu terhadap posisi pada koordinat (x, y) :



Gambar XI: *Bezier Curve* pada *keyframing*^[9]

Sumbu x pada gambar di atas menyatakan waktu sedangkan sumbu y menyatakan posisi. Simpul $P_1 = (0,0)$ dan $P_2 = (1,1)$. Simpul berwarna merah dan biru menyatakan *control points* yang masing-masing memiliki koordinat $P_2 = (0.2, 0.5)$ dan $P_3 = (0.5, 0.9)$. Untuk menghitung posisi, dapat digunakan rumus umum sebagai berikut:

$$P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4, \text{ untuk } 0 \leq t \leq 1 \quad (6)$$

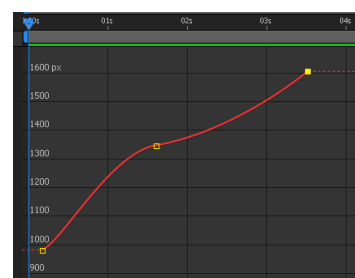
t pada rumus di atas tidak menyatakan waktu, melainkan sebuah representasi parametris. Untuk mengetahui visualisasi t , carilah panjang kurva tersebut, lalu kalikan dengan nilai t . Metode ini seperti meluruskan kurva tersebut, lalu mengalikannya dengan nilai t . Misalkan nilai $t = 0.5$, maka t adalah 0.5 dikali panjang kurva. Panjang kurva tersebut dapat ditemukan dengan menggunakan kalkulus, yang tidak dibahas pada bagian ini.

Perhitungan nilai pada titik koordinat x pada gambar XI pada $t = 0.5$ menggunakan persamaan (6) yaitu:

$$x(0.5) = (1-0.5)^3 \times 0 + 3 \times 0.5(1-0.5)^2 \times 0.2 + 3 \times (0.5)^2(1-0.5) \times 0.5 + (0.5)^3 \times 1$$

didapatlah $x(0.5) = 0.3875$. Nilai pada titik koordinat $y(0.5)$ menggunakan persamaan (6) adalah 0.65. Maka dapat disimpulkan, pada waktu = 0.3875 satuan, didapat posisi = 0.65 satuan. Artinya, pada waktu awal animasi bergerak cepat terlebih dahulu namun perlahan-lahan bergerak lebih lambat. Ini membuat gerakan pada interpolasi *Spline* menarik dan lebih realistis dibandingkan interpolasi linier.

Pemakaian *Bezier Curve* pada program pembuat animasi Adobe After Effects adalah sebagai berikut:



Gambar XII: *Bezier Curve* pada Adobe After Effects

Pada gambar XII, terdapat tiga simpul. Sebenarnya, terdapat

$3+2+2$ simpul = 7 simpul, namun 4 simpul yang merepresentasikan *control points* tersebut tidak diperlihatkan pada program ini. Pada gambar XII, diperlihatkan bahwa kecepatan transisi dari simpul paling kiri ke simpul tengah hampir konstan, namun bergerak lebih lambat sedikit beberapa saat sebelum mencapai simpul tengah. Dari simpul tengah ke simpul paling kanan, kecepatan transisi awalnya agak landai, lalu perlahan-lahan dipercepat.

IV. KESIMPULAN

Teknik interpolasi pada *keyframing* dapat membuat *frame-frame* baru dengan hanya mengetahui beberapa bingkai kunci saja. Dengan interpolasi lanjar dan interpolasi *Spline* yang dibahas pada makalah ini, kecepatan gerakan yang dihasilkan bisa berbeda. Interpolasi lanjar menghasilkan gerakan yang konstan sedangkan interpolasi *Spline* menghasilkan gerakan yang tidak konstan tetapi variatif, membuat gerakan semakin menarik. Adapun rumus untuk menghitung kecepatan rata-rata *frame* dan mengetahui pada posisi manakah gambar sekarang sangat berguna untuk animator dan *programmer* pembuat aplikasi animasi. Dalam interpolasi yang lebih kompleks, suatu *keyframe* bisa memiliki banyak sekali objek yang dianimasikan, namun intelegensia buatan bisa memprediksi gerakan animasi yang dihasilkan dengan menggunakan interpolasi.

V. UCAPAN TERIMAKASIH

Penulis ingin berterimakasih kepada Dr. Rinaldi Munir, M.T. sebagai dosen Matematika Diskrit kelas 01 yang telah membuka wawasan saya mengenai dunia matematika diskrit bukan hanya pada aspek teori saja, namun banyak sekali aplikasi matematika diskrit pada kehidupan sehari-hari. Selain itu, penulis berterimakasih kepada semua penulis referensi sehingga makalah ini dapat terselesaikan. Tidak lupa penulis berterimakasih pada keluarga dan teman-teman yang sudah membantu penulis dalam pengerjaan makalah ini, beserta Tuhan Yang Maha Esa yang sudah memberi hikmat pada penulis.

REFERENCES

- [1] Munir, Rinaldi, Matematika Diskrit, Bandung: Informatika Bandung, 2009.
- [2] <http://mathworld.wolfram.com/SimpleGraph.html/> diakses pada 6 Desember 2018.
- [3] <http://www.catatanrobert.com/relasi-dan-representasinya-dengan-tabel-matriks-dan-graf-berarah/> diakses pada 6 Desember 2018.
- [4] <http://mathworld.wolfram.com/Interpolation.html/> diakses pada 7 Desember 2018.
- [5] <https://www.khanacademy.org/partner-content/pixar/animate/ball/pi/animation-with-linear-interpolation/> diakses pada 7 Desember 2018.
- [6] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-837-computer-graphics-fall-2012/lecture-notes/MIT6_837F12_Lec01.pdf/ diakses pada 7 Desember 2018.
- [7] <http://web.arch.virginia.edu/arch545/handouts/keyframing.html/> diakses pada 7 Desember 2018.
- [8] Kovar, Lukas; Gleicher, Michael; Pighin, Frederic, Motion Graphs, SIGGRAPH '02: 29th International Conference on Computer Graphics and Interactive Techniques.
- [9] <http://cubic-bezier.com/#.2,.5,.5,.9/> diakses pada 9 Desember 2018.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017



Edward Alexander Jaya
13517115