

Simulasi Animasi dengan Menggunakan Pendekatan Diferensial

Iping Supriana Suwardi^{1*}, Wanda Gusdya Purnama², Dicky Prima Satya³, Edvin Ramadhan⁴, Fitriyani⁵

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
email: ^{1*}iping@informatika.org
²wanda.gusdya@unpas.ac.id
³dicky@informatika.org
⁴edvin.ramadhan@students.itb.ac.id
⁵fitriyani@telkomuniversity.ac.id
Jl. Ganeca No 10, Bandung 40132

Abstrak

Dalam pemutaran animasi, awalnya dibutuhkan banyak sekali gambar. Misalkan untuk animasi dengan kualitas *24 frames per second*, dibutuhkan 24 gambar untuk setiap detiknya. Artinya, jika animasi yang dibuat berdurasi 10 menit, dibutuhkan 14.400 gambar. Padahal, satu gambar dengan gambar lainnya tidak begitu berbeda. Perkembangan teknologi memungkinkan gambar-gambar yang sama atau mirip untuk dibangkitkan oleh komputer. Hal tersebut dapat memberikan efisiensi bagi proses pembuatan animasi. Makalah ini bertujuan untuk mengusulkan salah satu metode pembuatan animasi yang efisien, yang disebut metode pembuatan animasi dengan pendekatan diferensial. Metode pembuatan animasi dibagi menjadi empat langkah, yaitu analisis *keyframe*, penentuan *key location* untuk setiap *keyframe*, komputasi interpolasi, dan pembangkitan *frame* antara. Hasil pengujian metode ini menunjukkan bahwa *frame* yang digunakan dalam sebuah tayangan dapat berkurang 96,25%.

Kata kunci: animasi, simulasi, diferensial, *tweening*

1 PENDAHULUAN

Animasi merupakan salah satu teknik penyampaian informasi yang efektif, karena sistem visual manusia adalah pemroses informasi yang canggih yang dirancang untuk dapat melihat dan menerjemahkan pergerakan. Ketika sebuah animasi dibuat, biasanya animasi tersebut direkam dalam sebuah film atau video sebagai kumpulan gambar yang ketika ditampilkan dengan cepat akan terlihat sebagai sebuah gambar bergerak. Hal tersebut dimungkinkan karena mata dan otak manusia menyusun gambar-gambar tersebut dan menerjemahkannya menjadi sebuah gerakan yang terus menerus. Sebuah gambar yang disajikan untuk seseorang dalam waktu yang singkat akan meninggalkan jejak yang dinamakan *positive afterimage* pada sistem visualnya untuk waktu yang singkat setelah gambar tersebut menghilang (De Lauretis and Heath, 1980). Fenomena tersebut

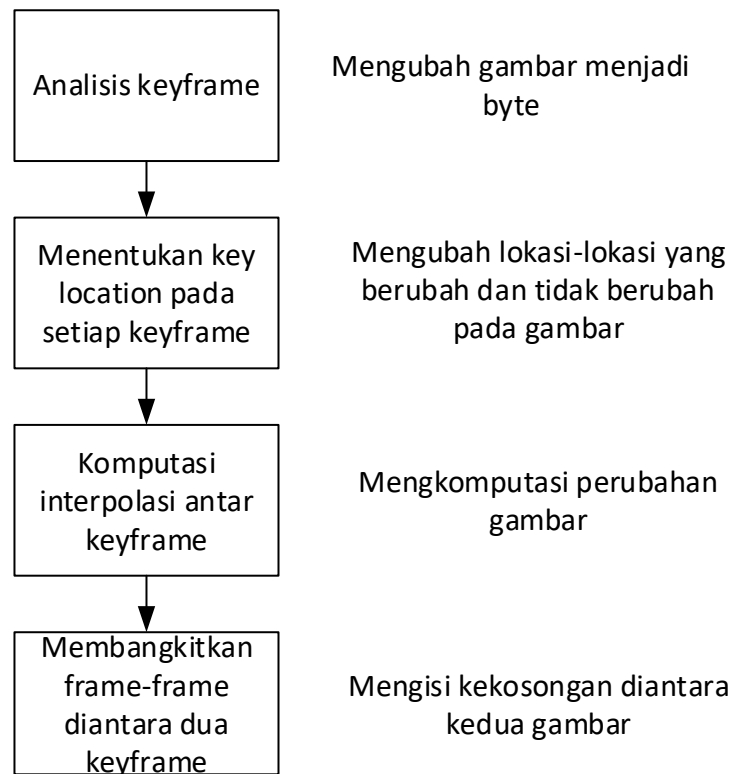
dinamakan *persistence of vision*. *Persistence of vision* bisa didefinisikan sebagai fenomena ketika sebuah film yang terdiri dari gambar-gambar statis dipertontonkan dengan kecepatan yang konstan, gambar statis yang tersebut akan dicampur oleh mata manusia sehingga menghasilkan ilusi gambar bergerak (Giannetti, 2013). Hal tersebut juga menyebabkan jika seseorang melihat gambar-gambar diam yang saling berkaitan secara terus menerus, maka ia akan merasakan sensasi sebuah gambar yang bergerak. *Positive afterimage* dari setiap gambar mengisi kekosongan diantara gambar-gambar. Fenomena tersebut sering digunakan dalam teknik animasi untuk menentukan jumlah gambar dalam satu detik animasi (Nojiri *et al.*, 2014).

Untuk menghasilkan animasi dalam skala yang besar, awalnya diperlukan gambar yang sangat banyak. Hal ini tentu saja menyebabkan sumber daya yang diperlukan untuk membuat sebuah animasi skala besar menjadi sangat banyak. Komputer memungkinkan efisiensi untuk Teknik animasi modern. Bidang animasi komputer mengawinkan seni dan perancangan kreatif dengan ilmu matematika dan komputer (Liang *et al.*, 2016). Teknik animasi modern tidak lagi menggunakan gambar sebanyak *frame* yang dibutuhkan, namun memanfaatkan *key frame*. *Key frame* adalah *frame* yang dapat mewakili konten dan informasi yang menonjol dari sebuah bidikan gambar (Algur and Vivek, 2016). *Key frame* menentukan poin awal dan akhir sebuah animasi, yang akan diproses oleh komputer menjadi transisi yang halus. *Key frame* memungkinkan animasi untuk disimpan dalam *file* dengan ukuran yang lebih kecil, karena yang disimpan hanya sekumpulan *frame* kunci. Proses penentuan *key frame* menjadi kunci bagi optimasi efisiensi bagi penyimpanan *file* animasi (video) dan pemrosesannya saat diputar.

Semakin banyak *frame* atau *key frame* yang disimpan, maka ukuran *file* video yang disimpan akan semakin besar. Ukuran video yang besar akan memerlukan media penyimpanan yang besar dan memakan waktu yang lama untuk pengiriman data pada jejaring komputer. Lamanya waktu yang dibutuhkan untuk mengirimkan data video melalui jejaring komputer menjadi persoalan utama saat ini, seiring meningkatnya kebutuhan akan konten video di internet terutama dengan semakin populernya media sosial. Berbagai teknik dan algoritma yang digunakan untuk menyimpan gambar bergerak menerapkan kompresi berdasarkan *key frame*. Kompresi yang dilakukan dapat mengurangi kebutuhan penyimpanan data secara signifikan yang juga berakibat pada pengurangan waktu pengiriman data dalam jaringan.

2 METODOLOGI PENELITIAN

Penelitian ini menggunakan metodologi dengan empat langkah, yaitu analisis *keyframe*, penentuan *key location* pada setiap *frame*, komputasi interpolasi antar *frame*, dan pembangkitan *frame-frame* di antara dua *frame*, seperti diperlihatkan pada Gambar 1.



Gambar 1: Metodologi Penelitian

2.1 Analisis *keyframe*

Analisis *keyframe* dilakukan terhadap sebuah video untuk mendapatkan *frame-frame* kunci. Setiap *frame* dari video diperiksa tingkat kemiripannya dengan *frame* tetangganya. Pemeriksaan kemiripan dilakukan dengan cara mengubah *frame* menjadi matriks warna, lalu menghitung perbedaan warna dari setiap koordinat matriks. *Frame* yang mirip dengan *frame* sebelumnya, tidak akan dipilih menjadi *keyframe* untuk melakukan efisiensi dalam proses pemutaran animasi. *Threshold* yang ditetapkan pada tahap ini adalah 80%, artinya hanya dua *frame* yang tingkat kemiripan warnanya dibawah 80% yang akan disimpan. Rumus yang digunakan untuk menghitung perbedaan warna yaitu *Euclidean distance* (Gibson, 2003), di mana:

$$|Z| = \sqrt{x^2 + y^2} \quad (1)$$

Sehingga

$$distance = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \quad (2)$$

2.2 Penentuan *key location* pada setiap *frame*

Keyframe yang didapatkan pada tahapan sebelumnya, akan disimpan dengan posisi *frame* pada video. *Frame* tersebut kemudian akan dibandingkan dengan *frame* selanjutnya, untuk ditentukan titik-titik kunci yang merupakan titik koordinat pada matriks *image* dari *frame* yang warnanya berbeda dengan titik koordinat pada matriks *image frame* sebelumnya. *Key location* tersebut akan disimpan posisinya untuk kemudian dibuat interpolasinya, sejumlah *gap* di antara dua *frame*. Rumus yang digunakan untuk menghitung perbedaan dua titik tersebut sama dengan rumus *distance* yang digunakan pada tahapan analisis *keyframe*, dengan nilai *threshold* kesamaan warna 1.000.

2.3 Komputasi interpolasi antar *frame*

Titik interpolasi dibangkitkan dengan membaurkan dua warna yang berbeda, yang

didapatkan dari dua titik koordinat yang sama pada dua *keyframe* yang berbeda. Interpolasi artinya data baru yang diestimasi *diantara* dua nilai yang diketahui (*Interpolate / Definition of Interpolate by Merriam-Webster, 2018*). Esensi dari interpolasi adalah untuk merepresentasikan fungsi yang selalu berubah-ubah sebagai fungsi penjumlahan diskrit yang berbobot dan bergerak (Thévenaz, Blu and Unser, 2000). Interpolasi pada konteks animasi merupakan gambar *diantara* dua gambar yang berbeda, yang akan menghasilkan ilusi bergerak. Rumus yang digunakan untuk membangun data interpolasi adalah sebagai berikut. Dimana *blending* merupakan tingkat intensitas pembauran warna (0 – 1).

$$R_{blend} = (R_1 \times blending) + (R_2 \times blending') \quad (2)$$

$$G_{blend} = (G_1 \times blending) + (G_2 \times blending') \quad (3)$$

$$B_{blend} = (B_1 \times blending) + (B_2 \times blending') \quad (4)$$

2.4 Pembangkitan *frame-frame* di antara dua *frame*

Setelah data interpolasi didapatkan, barulah data tersebut disimpan menjadi *frame-frame* baru. Hanya titik yang berbeda yang akan disimpan, sedangkan titik dengan warna yang sama tidak akan disimpan, dan akan menggunakan data *keyframe* sebelumnya ketika pemutaran animasi. Hal tersebut dilakukan untuk melakukan efisiensi penyimpanan data. *Frame-frame* tersebut disimpan dalam format rangkaian bit fxyrgb. Dimana f adalah nomor frame, x adalah koordinat x, y adalah koordinat y, r adalah intensitas warna merah, g adalah intensitas warna hijau, dan b adalah intensitas warna biru. Contoh satu *record* dalam penyimpanan adalah 00000100010001255255000, yang berarti itu adalah titik pada frame ke-1, dengan koordinat (1,1), dengan intensitas warna merah 255, hijau 255, dan biru 0.

3 HASIL DAN PEMBAHASAN

Simulasi yang dilakukan menggunakan sebuah video berdurasi 19 menit dan 40 detik, dengan 35.400 *frame*. Proses pemilihan *keyframe* menghasilkan sejumlah 1.328 *keyframe* yang disimpan. Hanya 3,75% dari total keseluruhan frame yang betul-betul berbeda dan layak untuk disimpan.



a



b



c

Gambar 2: Contoh *Keyframe* yang dipilih

Jumlah *frame* yang dibangkitkan untuk setiap *gap* bervariasi, bergantung pada lebar *gap* yang dimiliki oleh dua *keyframe* tersebut. Untuk pasangan *keyframe* a pada Gambar 2 contohnya, dibangkitkan 9 *frame*.



Gambar 3: Hasil pembangkitan *frame*

4 KESIMPULAN

Metode yang diusulkan pada makalah ini telah dapat mengurangi jumlah frame dari sebuah video berdurasi 19 menit dan 40 detik. Dari total 35.400 frame, disusutkan menjadi 1.328 frame. Efisiensi penggunaan frame dapat dilakukan, karena pemutaran animasi hanya membutuhkan 3,75% dari total keseluruhan frame. Pemutaran animasi dengan 3,75% frame utama tersebut menghasilkan kualitas animasi yang dapat diterima. Penggunaan audio belum disertakan dalam penelitian ini, dan dapat diusulkan untuk penelitian berikutnya.

Referensi

- Algur, S. P. and Vivek, R. (2016) 'Video Key-Frame Extraction using Entropy value as Global and Local Feature', *arXiv preprint arXiv:1605.08857*, pp. 1–5.
- Giannetti, L. D. (2013) *Understanding Movies*. Boston: Pearson.
- Gibson, C. G. (2003) *Elementary Euclidean Geometry an Introduction*. New York: Cambridge University Press.
- Interpolate* / *Definition of Interpolate by Merriam-Webster* (2018) *Merriam-Webster Dictionary*.
- De Lauretis, T. and Heath, S. (1980) *The Cinematic Apparatus*. London: The Macmillan Press Ltd.
- Liang, H. *et al.* (2016) 'Computer animation data management: Review of evolution phases and emerging issues', *International Journal of Information Management*. Elsevier Ltd, 36(6), pp. 1089–1100. doi: 10.1016/j.ijinfomgt.2016.07.008.
- Nojiri, K. *et al.* (2014) 'Present information through afterimage with eyes closed', *Proceedings of the 5th Augmented Human International Conference on - AH '14*, pp. 1–4. doi: 10.1145/2582051.2582054.
- Thévenaz, P., Blu, T. and Unser, M. (2000) 'Interpolation revisited', *IEEE Transactions on Medical Imaging*, 19(7), pp. 739–758. doi: 10.1109/42.875199.