

Pengaplikasian Aljabar Vektor dalam *Non-Photorealistic Rendering*

Muhammad Reifiza 13514103
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13514103<at>std.stei.itb.ac.id

Abstrak—*Non-Photorealistic Rendering* adalah sebuah teknik *rendering* yang tidak mengutamakan realisme untuk hasilnya. Teknik ini termotivasi dari teknik menggambar tradisional yang tidak mengutamakan realisme dalam hasilnya. Oleh sebab itu, beberapa hal baru harus dikenalkan seperti *stroke texture*. Dalam implementasinya, teknik ini menggunakan beberapa teori dan aljabar vektor di dalamnya. Dalam makalah ini penulis akan membahas beberapa metode NPR yang digunakan saat ini dan menunjukkan implementasi aljabar vektor di dalamnya.

Keywords—Aljabar Vektor, Non-Photorealism, Rendering

I. PENDAHULUAN

Salah satu cabang dari bidang Teknik Informatika adalah grafika komputer dimana perhitungan-perhitungan dan analisis-analisis matematika dan fisika yang cukup runyam diimplementasikan dalam bentuk algoritma-algoritma yang juga runyam. Sudah banyak penerapan grafika komputer dalam kehidupan sehari-hari, contohnya ada dalam bidang publikasi digital, dimana banyak gambar yang dihasilkan sebenarnya merupakan olah citra komputer. Contoh lain adalah dalam bidang seni digital. Banyak karya yang dihasilkan oleh *digital artist* (seniman digital) tidak mungkin dihasilkan tanpa bantuan komputer. Sebabnya bermacam-macam, misalnya untuk mensimulasikan gambar yang tidak mungkin diambil/ada di dunia nyata.

Sudah banyak penelitian ilmiah yang bertujuan untuk mengembangkan bidang grafika komputer. Namun, kebanyakan dari penelitian itu bertujuan untuk menghasilkan gambar yang se-realistis seperti foto sebisa mungkin (*photorealism*). Banyak teknik yang digunakan untuk *me-render* model tiga dimensi supaya persis dengan kehidupan nyata. *Rendering* seperti ini umumnya disebut *Photorealistic Rendering*.

Gambar yang dihasilkan dengan *photorealistic rendering* tidak memiliki garis pinggir atau garis kontur yang jelas untuk memberikan kesan tiga dimensi, tetapi menggunakan tekstur dan warna gelap-terang, mirip seperti di dunia nyata. Kenyataannya, tidak selalu gambar seperti ini merupakan yang terbaik. Gambar yang menampilkan garis pinggir, garis kontur, atau garis lainnya yang tampil seperti buatan tangan lebih digunakan untuk

ilustrasi dalam buku pelajaran, buku manual, dan lain-lain yang membutuhkan *detail* yang tinggi, daripada gambar yang mirip seperti dunia nyata. Sebabnya, gambar yang mirip seperti dunia nyata memiliki *detail* yang cenderung dibiarkan oleh otak manusia, atau malah menjadi samar karena adanya tekstur yang saru dengan *detail* (misalnya pinggir objek).

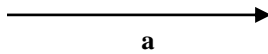
Membuat ilustrasi objek yang bagus tanpa meninggalkan *detail* objek dengan tangan dalam bentuk tiga dimensi merupakan tugas yang terkadang kurang menyenangkan, apalagi jika *detail*-nya sangat kompleks (misalnya mesin mobil). Walaupun dibantu dengan aplikasi pengolah grafis berbasis vektor, tetap saja tugas ini masih kurang menyenangkan. Akan lebih baik jika saja objek tersebut dapat dimodelkan dalam ruang tiga dimensi, lalu di-*render* dengan hasil yang tampak seperti buatan tangan. Sebuah teknik *rendering* dalam bidang grafika komputer, *Non-Photorealistic Rendering*, dikembangkan sedemikian rupa supaya dapat merealisasikan hal ini.

Teknik NPR merupakan teknik *rendering* yang tergolong baru, lebih muda daripada *Photorealistic Rendering*. Teknik ini mulai dikembangkan secara masif sekitar tahun 1994, ketika Winkenbach dan Salensin mempublikasikan karya mereka untuk menciptakan ilustrasi bergaya *Pen-and-Ink*[1]. Setelahnya, berbagai metode untuk menciptakan gambar yang bergaya kurang lebih mirip mulai bermunculan[3][4][7][8][9]. Secara garis besar, metode-metode tersebut menggunakan masukan pengguna sebagai parameter prosedur untuk menciptakan efek *brush stroke*, karena pada umumnya metode-metode ini menciptakan efek secara prosedural.

Beberapa jurnal mengkhususkan untuk mengevaluasi teknik NPR yang ada. Schaumann menemukan bahwa ilustrasi NPR, ilustrasi plot CAD, dan ilustrasi bergaya realisme menciptakan kesan yang berbeda bagi yang melihatnya[6]. Isenberg et.al. menemukan fakta bahwa ilustrasi NPR yang bergaya lebih realistis lebih disukai daripada ilustrasi NPR yang kurang realistis[5].

II. TEORI VEKTOR

Vektor adalah besaran yang memiliki nilai dan arah.



Gambar 1. Sebuah vektor **a**

Vektor **a** biasa dinotasikan dengan ‘**a**’ (gambar 1). Sebuah ruang vektor **V** adalah himpunan tidak kosong vektor-vektor yang terbentuk dari penjumlahan dan perkalian vektor dengan skalar[10]. Akibat dari hal ini, **V** selalu memenuhi aksioma berikut ini.

1. Penjumlahan **u** dan **v**, dimana **u** dan **v** adalah vektor di dalam **V**, dinotasikan dengan **u + v** selalu berada di dalam **V**.
2. **u + v = v + u**
3. **(u + v) + w = u + (v + w)**
4. Ada vektor **0** di **V** sehingga **u + 0 = u**
5. Ada vektor **-u** di **V** sehingga **u + (-u) = 0**
6. Perkalian vektor **u** di **V** dengan sebuah konstanta **k**, dinotasikan dengan **ku**, selalu ada di **V**.
7. **k(u + v) = ku + kv** dengan **k** adalah sebuah konstanta
8. **(k₁ + k₂)u = k₁u + k₂u** dengan **k₁** dan **k₂** adalah sebuah konstanta
9. **cd u = c(du) = (cd)u** untuk **c** dan **d** konstanta sembarang
10. **1u = u**

Hasil perkalian dalam (*dot product*) dari vektor **u** dan **v** yang berada di **V** dinotasikan dengan **u.v** dan berlaku

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \alpha \quad (1)$$

dengan α adalah besar sudut yang berada di antara vektor **u** dan **v**, dan $\|\mathbf{u}\|$ menyatakan panjang sebuah vektor. Jika

$$\mathbf{u} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \text{ dan } \mathbf{v} = \begin{pmatrix} d \\ e \\ f \end{pmatrix} \text{ maka } \|\mathbf{u}\| = \sqrt{a^2 + b^2 + c^2} \text{ dan}$$

$\|\mathbf{v}\| = \sqrt{d^2 + e^2 + f^2}$ dan hasil perkalian dalam dari **u** dan **v** juga dapat dihitung menggunakan

$$\mathbf{u} \cdot \mathbf{v} = ad + be + cf \quad (2)$$

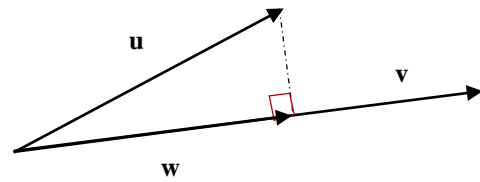
Jika **u**, **w** dan **v** berada di dalam \mathcal{R}^n , dan **k** adalah sebuah skalar, maka

1. **u.v = v.u**
2. **k(u.v) = (ku).v = u.(kv)**
3. **(u + v).w = u.w + v.w**
4. **u.u = 0** jika dan hanya jika **u = 0**

Jika **u** dan **v** berada di dalam \mathcal{R}^n , maka $\text{proj}_{\mathbf{v}} \mathbf{u}$ adalah proyeksi ortogonal vektor **u** ke vektor **v**. Misalkan **w** adalah $\text{proj}_{\mathbf{v}} \mathbf{u}$, maka **w** berada di \mathcal{R}^n , maka $\|\mathbf{w}\| = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|}$ (3). Karena

$$\mathbf{1} = \mathbf{v} / \|\mathbf{v}\|, \text{ dengan } \mathbf{1} \text{ adalah vektor } \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \text{ maka } \mathbf{w} =$$

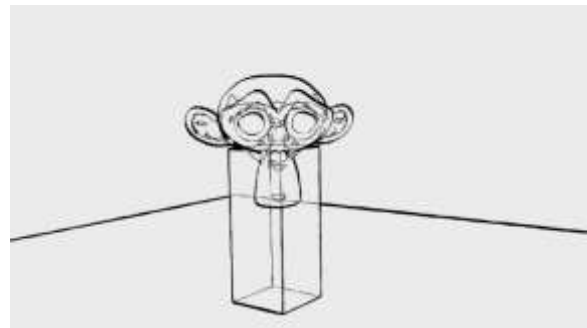
$$\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v} \quad (4).$$



Gambar 2. Proyeksi **u** ke **v** menghasilkan **w**

III. PRINSIP NPR

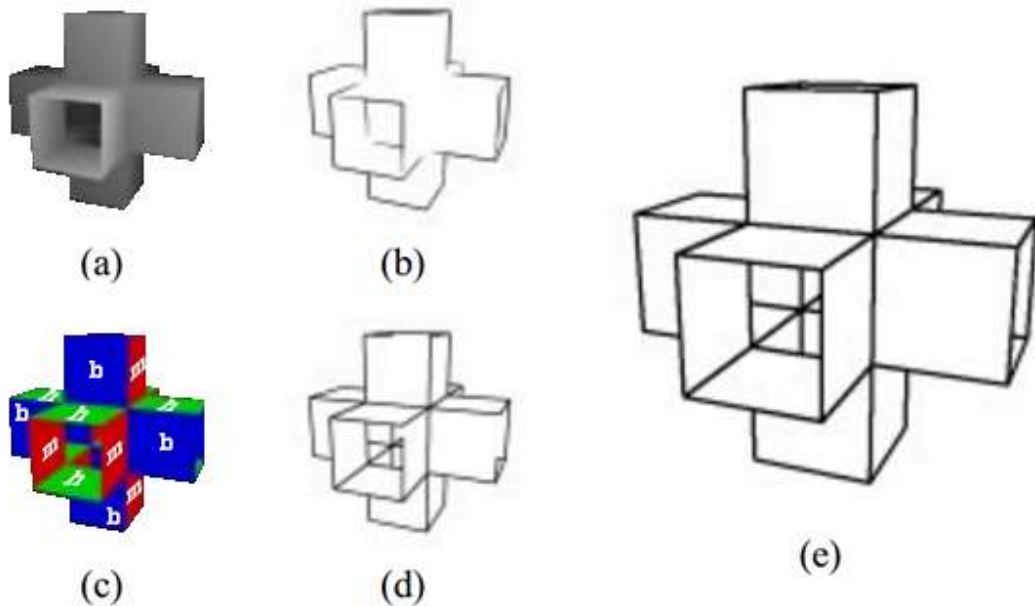
Teknik NPR dikembangkan berdasarkan teknik menggambar tradisional yang tidak beraliran realisme (tidak persis dengan dunia nyata). Ada banyak gaya yang dihasilkan oleh berbagai metode NPR. Salah satunya yang paling populer adalah gaya/teknik *pen-and-ink* (untuk selanjutnya disingkat menjadi PI), walaupun gaya-gaya yang lain seperti lukisan cat air adalah mungkin[7].



Gambar 3. Contoh sederhana penggunaan NPR dengan gaya *pen-and-ink*. Gambar dihasilkan melalui mode *freestyle* pada *open-source software* Blender.

Gambar yang dihasilkan dengan PI tidak pernah berwarna, jadi penggunaan garis (*strokes*) haruslah seoptimal mungkin untuk mengomunikasikan data objek (kontur, tekstur, dan hal-hal akibat adanya pencahayaan seperti refleksi, *volume scattering*, dan lain-lain) kepada pengamat (yang melihat gambar) secara efektif[1]. Karena gambar yang dihasilkan merupakan olah citra komputer, maka beberapa hal harus diperhatikan:

1. Garis yang tercipta di gambar haruslah terlihat senatural mungkin. Efek natural ini mungkin didapatkan dari pengacakan ketebalan garis, karena garis yang tebalnya sama rata akan terlihat terlalu mekanis dan tidak natural.
2. Garis yang terlalu tipis atau terlalu tebal dapat merusak gambar secara keseluruhan[1][7].
3. Impresi tentang warna dan kontur haruslah didapat dari kombinasi garis (*strokes*) yang ada, namun kombinasi garis tidak harus mencerminkan warna dan kontur secara akurat.[1][7]



Gambar 4. Tahapan *edge detecting* dengan metode *Depth Map* dan *Normal Map*. [2] Keterangan: b = biru, m = merah, h = hijau

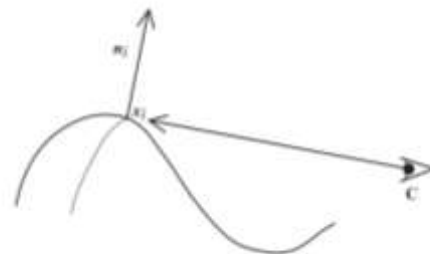
Secara garis besar, NPR bekerja dengan cara:

1. *Render* terlebih dahulu objek dengan normal
2. Lakukan *edge detecting* (deteksi sisi-sisi objek)
3. Gambar garis berdasarkan sisi-sisi objek.

Ada beberapa metode *edge detecting* yang tersedia saat ini [1] dan berbagai perangkat lunak menggunakan berbagai metode yang berbeda (atau mungkin kombinasi). Beberapa yang paling umum adalah penggunaan metode *Depth Map* (gambar 4(a), 4(b)) dan metode *Normal Map* (gambar(4c), 4(d)).

Pendekatan lainnya adalah menggunakan kurva *Silhouette* (siluet)[2]. Kurva siluet didefinisikan sebagai semua sisi pada sebuah muka yang menghubungkan muka hadap depan dengan muka hadap belakang untuk *polygonal meshes* (artinya *mesh/objek* tidak memiliki permukaan yang kontinu / *smooth*), atau vektor titik permukaan \mathbf{x}_i dengan titik normal \mathbf{N}_i yang tegak lurus dengan vektor penglihatan $\mathbf{N}_i - \mathbf{C}$ sehingga memenuhi persamaan

$$\mathbf{x}_i \cdot (\mathbf{N}_i - \mathbf{C}) = 0 \quad (5)$$



Gambar 5. Siluet untuk permukaan yang kontinu[2]

Cara penentuan lokasi titik-titik pembangun siluet adalah dengan melakukan interpolasi linear.

$$\mathbf{x} = \frac{\|\mathbf{d}_i\|}{\|\mathbf{d}_i\| + \|\mathbf{d}_j\|} \mathbf{x}_i + \frac{\|\mathbf{d}_j\|}{\|\mathbf{d}_i\| + \|\mathbf{d}_j\|} \mathbf{x}_j \quad (6)$$

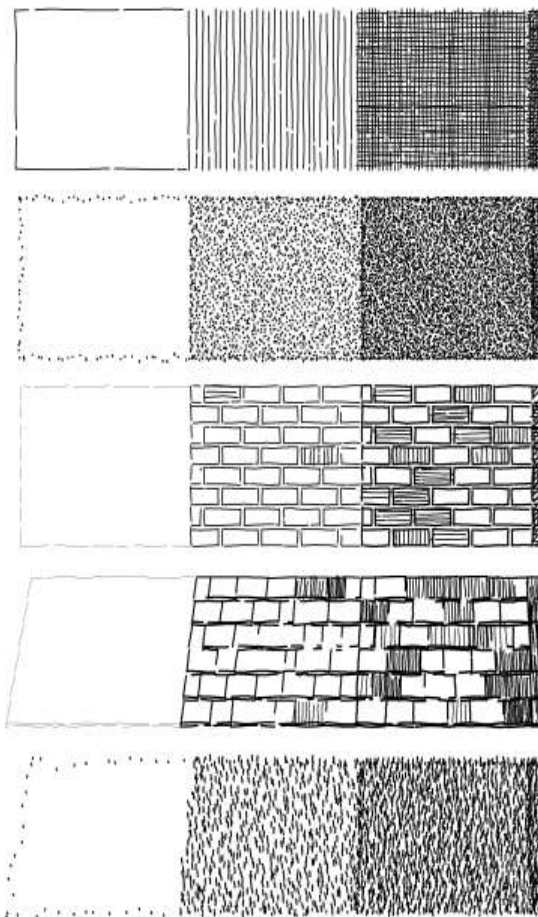
dimana \mathbf{d}_i merupakan vektor satuan dari hasil kali dalam (*dot product*) antara vektor titik permukaan \mathbf{x}_i dengan atau vektor penglihatan $\mathbf{N}_i - \mathbf{C}$.

Definisi pada persamaan (5) tidak menyatakan visibilitas dari kurva[2]. Oleh karena itu, tahapan selanjutnya adalah menentukan mana kurva yang dapat dilihat oleh kamera, mana yang tidak. Beberapa algoritma [8][9][12] mendeskripsikan pendekatan untuk hal ini.

Gambar yang dihasilkan dari metode-metode di atas adalah cukup jika hanya ingin pinggir objek yang ditampilkan sebagai goresan-goresan garis. Namun, jika goresan-goresan garis yang ditampilkan ingin

memunculkan kesan yang lebih natural (ketebalan yang bervariasi, goresan garis bertekstur, dan sebagainya) atau bahkan mewarnai objek yang tidak berbasis fisis (asalkan mampu memunculkan kesan tiga dimensi), maka beberapa tahapan lagi masih dibutuhkan untuk mencapainya.

Salah satu cara untuk memberikan variasi terhadap goresan adalah dengan memberikan *pressure function* dan *waviness function* [1]. Namun, merupakan suatu kemungkinan suatu goresan memiliki berbagai macam karakter yang bisa dicapai dengan tidak hanya kedua fungsi tadi. Selanjutnya, metode yang dikembangkan awalnya oleh Winkenbach dan Salesin ini dikembangkan lebih lanjut dengan menambahkan banyak fungsi lainnya yang bisa jadi memengaruhi karakter dari sebuah goresan [3][7][9].



Gambar 6. Beberapa contoh goresan-goresan yang digunakan Winkenbach et.al. untuk memberikan impresi tekstur pada PI [1]

Jika sebuah objek memiliki warna yang homogen, pada PI warna tersebut diilustrasikan dengan goresan-goresan yang sama rata. Lain ceritanya jika warnanya tidak homogen (seperti adanya efek pencahayaan tertentu), maka warna tersebut diilustrasikan dengan teknik yang mirip seperti pada gaya pointilis; semakin gelap maka semakin rapat goresan-goresannya [7]. Kemungkinan lain adalah dengan mengganti penampilan warna tersebut. [4][7][13]. Ada berbagai cara, misalnya penggantian dari warna yang

berasal dari *physically-based shader* menjadi warna yang lebih kartun.

Sebuah objek juga mungkin memiliki tekstur yang diterapkan padanya. Pada *pipeline* grafika komputer tradisional, tekstur yang diterapkan dalam sebuah objek umumnya merupakan sebuah bidang gambar dua dimensi. Teknik NPR sedikit mengubah hal ini dengan mengganti tekstur berbasis gambar menjadi tekstur berbasis goresan sehingga gambar mungkin menjadi berubah warnanya (dalam hal bukan PI) [1][7].

Sebagai contoh, gambar 6 merupakan tekstur yang digunakan Winkenbach dan Salesin dalam metode NPR miliknya[1]. Curtis et.al. memiliki pendekatan yang mirip dengan Winkenbach dalam hal penggunaan goresan tekstur untuk mengubah gambar yang realis menjadi seperti lukisan minyak[7]. Garis besar dari metode NPR yang diterapkan oleh Curtis adalah sebagai berikut:

1. Gambar disamarkan hingga ke level tertentu
2. Tambahkan *detail* yang hilang akibat penyamaran (blurring) dengan menggunakan tekstur goresan

Sementara itu, Gooch et.al. memberikan sebuah metode yang memungkinkan untuk mewarnai objek dalam tiga dimensi secara *non-photorealistic* [4]. Metode ini dikembangkan dengan motivasi pengilustrasian objek-objek atau hal-hal teknis (misalnya mesin, gerinda, dan sebagainya) yang memiliki artian goresan sedikit lain daripada ilustrasi artistik[15] (metode-metode yang dibahas sebelumnya umumnya digunakan untuk ilustrasi artistik). Goresan tebal digunakan untuk menggariskan tepian objek. Variasi garis yang digunakan adalah garis yang berangsur-berangsur menipis ketika mencapai titik terjauh objek yang dilihat kamera (berarti garis yang paling tebal dalam hal ini adalah garis ‘terdepan’). Warna garis paling umum digunakan adalah warna hitam, seperti kebanyakan NPR[15].

Konvensi pewarnaan untuk ilustrasi teknis juga agak berbeda dengan pewarnaan untuk ilustrasi artistik[15]. Umumnya warna yang digunakan adalah warna putih untuk warna *diffuse* dan warna yang sangat bergradasi jika objek terbuat dari bahan metalik, seakan-akan objek sangat *anisotropis*[15][4]. Penggunaan gelap-terang sangat memengaruhi orientasi benda terhadap kamera dan impresi kontur yang tercipta[15].

Biasanya, pewarnaan untuk material *diffuse* dihitung dengan persamaan

$$I = k_d k_a + k_d \max(0, \mathbf{l} \cdot \mathbf{n}) \quad (7)$$

dengan I adalah warna RGB (*red-green-blue*) yang akan ditampilkan untuk sebuah titik pada permukaan, k_d adalah koefisien reflektansi RGB *diffuse* pada titik tersebut, k_a adalah koefisien RGB *ambient illumination*, \mathbf{l} adalah vektor satuan cahaya (vektor arah datangnya cahaya), dan \mathbf{n} adalah vektor satuan permukaan. Masalah umum dengan persamaan ini adalah untuk suatu k_a dan k_d , gambar yang diperoleh kehilangan banyak detail, baik di area bayangan maupun area terang [4]. Masalah ini dapat diselesaikan dengan menggeneralkan persamaan di atas menjadi

persemaan yang menggunakan variabel k_{cool} dan k_{warm} untuk merepresentasikan warna ber'temperatur' dingin (biru, ungu, hijau) dan warna ber'temperatur' hangat (merah, jingga, kuning) [4].

$$I = \left(\frac{1 + \mathbf{l} \cdot \mathbf{n}}{2}\right) k_{cool} + \left(1 - \frac{1 + \mathbf{l} \cdot \mathbf{n}}{2}\right) k_{warm} \quad (8)$$

IV. KESIMPULAN

Non-Photorealistic Rendering telah terlihat banyak perkembangan sejak pertengahan 90-an. Banyak metode yang telah dikembangkan yang juga berbeda-beda dalam segi gaya artistik yang dibawahnya. Dilihat dari segi desainer dan *digital artist*, NPR adalah peralatan baru yang dapat digunakan untuk menuangkan kreatifitas dan memudahkan untuk membuat ilustrasi objek-objek dengan detail yang rumit, karena NPR memang diciptakan untuk mempermudah pengilustrasian detail objek. Walaupun metode NPR yang ada sudah lumayan banyak dan berbeda-beda, umumnya prosedur NPR yang ada dipengaruhi juga oleh masukan pengguna sebagai parametrisasi (interaktif).

V. UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Kuasa karena berkat rahmat-Nya penulis dapat menyelesaikan makalah ini. Terima kasih kepada Bapak Rinaldi Munir dan Bapak Judhi Santoso sebagai dosen mata kuliah Aljabar Geometri. Semua metode yang dibahas di sini bukan milik penulis, namun kepada pemegang hak ciptanya masing-masing, yang telah banyak diaplikasikan sehari-hari.

REFERENCES

- [1] Winkenbach, Georges dan David H. Salesin, "Computer-Generated Pen-and-Ink Illustration". Department of Computer Science and Engineering University of Washington, 1996.
- [2] Hertzmann, Aaron. 'Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines'. Diakses dari www.mrl.nyu.edu/hertzmann pada tanggal 14 Desember 2015.
- [3] Grabli, Stephane et.al, 'Programmable Style for NPR Line Drawing' dalam A. Keller (editor), *Eurographics Symposium on Rendering*. The Eurographics Association, 2004.
- [4] Gooch, Bruce et.al. 'A Non-Photorealistic Lighting Model For Automatic Technical Illustration' dalam Michael Cohen (editor), *SIGGRAPH 98 Conference Proceedings*, 1998.
- [5] Isenberg, Tobias et.al. 'Non-Photorealistic Rendering in Context: An Observational Study', 2006.
- [6] Schaumann, J. et.al. 'Assessing the Effect of Non-photorealistic Rendered Images in CAD'. Dalam *Proceedings of CHI'96*. ACM Press, 1996.

- [7] Cassidy J. Curtis, Sean E. Anderson, Kurt W. Fleischer, dan David H. Salesin. 'Computer-Generated Watercolor'. dalam *SIGGRAPH 97 Conference Proceedings*, 1997.
- [8] Elber, Gershon dan Elaine Cohen. 'Hidden Curve Removal for Free-Form Surfaces'. In *SIGGRAPH 90 Conference Proceedings*, 1990.
- [9] Markosian, L. et.al. 'Real-Time Non-Photorealistic Rendering', dalam *SIGGRAPH 97 Conference Proceedings*, 1997.
- [10] Lay, David C. "Linear Algebra and Its Application 4th Edition.". Addison Wesley, 2012.
- [11] Lohan, Frank , "Pen and Ink Techniques". Contemporary Books, Inc., 1978.
- [12] Appel, Arthur , 'The Notion of Quantitative Invisibility and the Machine Rendering of Solids' dalam *Proceedings ACM National Conference*, 1967.
- [13] Gooch, B. dan Gooch, A. "Non-Photorealistic Rendering". AK Peters Ltd., 2001.
- [14] Martin, Judy, "Technical Illustration: Materials, Methods, and Techniques, vol. 1". Macdonald & Co Publishers, 1989.

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, 8 Desember 2015

ttd



Muhammad Reifiza 13514103