# Penggunaan Quaternion dan Matriks pada Perputaran Spasial

Jeremia Jason Lasiman 13514021[1]
*Program Studi Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
[1]*jeremia_jason@s.itb.ac.id*

***Abstrak*—Perputaran dapat direpresentasikan dengan beberapa cara. Cara yang umum digunakan adalah matriks perputaran, quaternion, dan sudut Euler. Diantara cara-cara tersebut, yang terbaik untuk digunakan adalah dengan representasi quaternion karena penggunaan memori yang kecil dan juga dapat menampilkan gerakan perputaran secara halus. Makalah ini akan membahas penggunaan matriks dan quaternion dalam perhitungan perputaran serta mengkaji mengapa quaternion menjadi representasi terbaik.**

***Kata kunci*—quaternion, matriks perputaran, *gimbal lock*, perputaran.**

## I. PENDAHULUAN

Perputaran tiga dimensi atau disebut perputaran spasial (*spatial rotation*) merupakan topik yang sangat sering dibahas. Aplikasi dari perputaran spasial ini juga sangat banyak dan dapat dipakai di berbagai bidang ilmu pengetahuan. Perhitungan mengenai perputaran spasial ini juga sangat berguna dalam bidang komputasi. Melalui perhitungan itu, dapat dilakukan simulasi mengenai suatu benda tidak hanya dari satu sudut, tetapi dari berbagai macam arah.

Perputaran spasial dalam matematika dapat dilakukan dengan beberapa cara. Dua cara umum yang digunakan untuk menghitung perputaran spasial adalah dengan menggunakan matriks dan quaternion. Namun, representasi quaternion lebih sering digunakan dalam perhitungan dibandingkan dengan representasi matriks.

## II. DASAR TEORI

### A. *Quaternion*

Quaternion adalah representasi tiga dimensi yang setara menggunakan bilangan kompleks. Quaternion ditemukan oleh William Rowan Hamilton pada tahun 1843.Quaternion dapat ditulis sebagai kombinasi linear dari elemen basis berupa $a1 + bi + cj + dk$, dimana *a*, *b*, *c*, and *d* are bilangan riil. Sedangkan *i*, *j*, *k* adalah unit imaginer yang mengikuti hukum Hamilton, yaitu

$$i^2 = j^2 = k^2 = ijk = -1 \quad (1)$$

$$ij = k;\ jk = i;\ ki = j;\ ji = -k;\ kj = -i;\ ik = -j \quad (2)$$

tetapi ketika aljabar vektor lebih ditekankan dari pada aljabar quaternion, maka *i*. *j*, *k* menjadi unit vektor Cartesian.

Operasi yang dapat dilakukan pada quaternion ada tiga, yaitu penjumlahan, perkalian skalar, dan perkalian quaternion.

1. *Penjumlahan Quaternion*

Dua quaternion $q_1$ dan $q_2$

$$q_1 = s_1 + ix_1 + jy_1 + kz_1 \quad (3)$$

$$q_1 = s_1 + ix_1 + jy_1 + kz_1 \quad (4)$$

Maka $q_1 + q_2$ adalah

$$q_1 + q_2 = 3 + i + j8 + k2. \quad (5)$$

2. *Perkalian Quaternion* (*Quaternion Product*)

Disebut juga produk Hamilton.
Dua quaternion $q_1$ dan $q_2$

$$q_1 = s_1 + \mathrm{v}_1 = s_1 + ix_1 + jy_1 + kz_1 \quad (6)$$

$$q_2 = s_2 + \mathrm{v}_2 = s_2 + ix_2 + jy_2 + kz_2 \quad (7)$$

maka hasil kalinya (*product*) adalah

$$q_1 q_2 = s_1 s_2 - \mathrm{v}_1 \cdot \mathrm{v}_2 + s_1 \mathrm{v}_2 + s_2 \mathrm{v}_1 + \mathrm{v}_1 \times \mathrm{v}_2 \quad (8)$$

Perkalian quaternion tidak bersifat komutatif, hal ini disebabkan hukum Hamilton yang menjadikan (2) dan perkalian tidak bersifat komutatif.

| × | 1 | i | j | k |
|---|---|---|---|---|
| 1 | 1 | i | j | k |
| i | i | −1 | k | −j |
| j | j | −k | −1 | i |
| k | k | j | −i | −1 |

Sumber : https://rooksheathscience.files.wordpress.com/2014/11/image96.png
**Gambar 1.1 Hasil kali quaternion**

3. *Besar Quaternion* (*Magnitude*)

Selain operasi penjumlahan dan perkalian, quaternion juga dapat dicari besarnya (*magnitude*).

Suatu quaternion

$$q = s + ix + jy + kz \tag{9}$$

Besarnya dinyatakan sebagai

$$||q|| = \sqrt{s^2 + x^2 + y^2 + z^2} \tag{10}$$

4. *Satuan Quaternion* (*Unit Quaternion*)

Seperti halnya vektor, quaternion juga memiliki bentuk satuan (*unit quaternion*). Satuan quaternion dapat dinyatakan sebagai

$$\hat{q} = \frac{q}{||q||} \tag{11}$$

5. *Quaternion Murni* (*Pure Quaternion*)

Suatu quaternion dapat disebut quaternion murni ketika skalarnya adalah 0. Misalkan

$$q_1 = ix_1 + jy_1 + kz_1 \text{ and } q_2 = ix_2 + jy_2 + kz_2 \tag{12}$$

6. *Konjugasi dari Quaternion*

Suatu quaternion

$$q = s + \mathrm{v}$$
$$q = s + ix + jy + kz \tag{13}$$

konjugasinya adalah

$$q = s - \mathrm{v} = s - (ix + jy + kz). \tag{14}$$

7. *Invers Quaternion*

Suatu quaternion

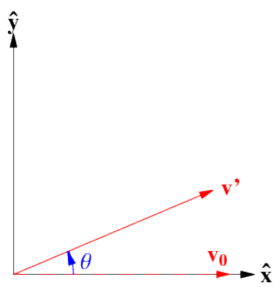
$$q = s + ix + jy + kz \tag{9}$$

*B. Matriks Perputaran*

Matriks perputaran (*rotation matrix)* adalah matriks yang digunakan untuk melakukan perputaran pada ruang Eucledian. Misalkan pada dimensi dua matriks

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{10}$$

memutar titik pada bidang Cartesian berlawanan arah jarum jam sesuai sudut $\theta$.




Sumber : [1]

**Gambar 2.1 Perputaran $v_0$ menjadi v'**

Pada gambar 2.1, $v_0$ dapat diputar menjadi v' dengan menggunakan (10) yang memiliki rumus perputaran

$$v' = R_0 \, v_0 \tag{11}$$

Dalam dimensi tiga, matriks perputaran memiliki tiga perputaran dasar. Perputaran dasar adalah perputaran yang hanya melibatkan satu sistem koordinat. Tiga perputaran dasar dengan sumbu *x, y*, dan *z*.

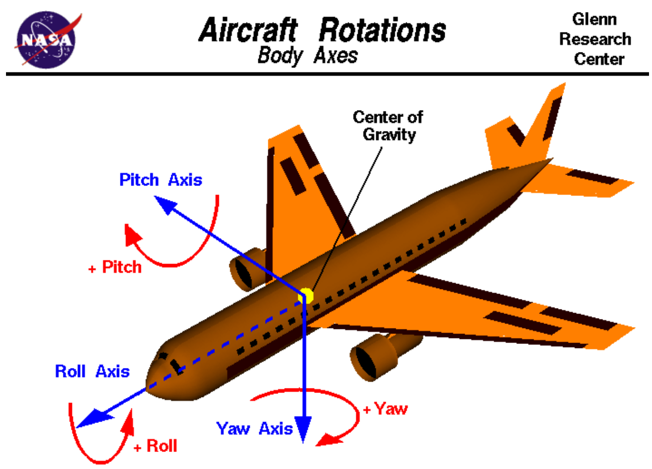
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{12}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{13}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Perputaran secara umum dapat diketahui dengan menghitung menggunakan perkalian dari matriks dasar. Misalkan pada sumbu yang biasa dikenal dalam istilah pembuatan pesawat, *yaw* $(\propto)$, *pitch* $(\beta)$, dan *roll* $(\gamma)$

$$R = R_z(\propto)R_z(\beta)R_z(\gamma) \tag{15}$$




Sumber : https://www.grc.nasa.gov/www/k-12/airplane/Images/rotations.gif

**Gambar 2.2 Yaw, Pitch, dan Roll rotasi pesawat**

## III. QUATERNION DAN MATRIKS DALAM PERPUTARAN

*A. Perputaran Quaternion*

Menurut teori perputaran Euler, pada dimensi tiga setiap perputaran atau perputaran yang beruntun terhadap suatu titik sama dengan sebuah perputaran dengan sudut $\theta$ terhadap suatu sumbu tertentu (disebut *sumbu Euler*) yang melewati suatu titik tertentu. Sumbu Euler tersebut biasa direpresentasikan sebagai vektor satuan **u**. Maka, setiap perputaran pada dimensi tiga dapat direpresentasikan sebagai kombinasi dari vektor **u** dan skalar $\theta$. Quaternion dapat dengan mudah merepresentasikan sumbu dan sudut.

Sebuah perputaran melalui sudut $\theta$ disekitar sumbu yang didefinisikan sebagai vektor satuan.

$$\vec{u} = (u_x, u_y, u_z) = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k} \qquad (16)$$

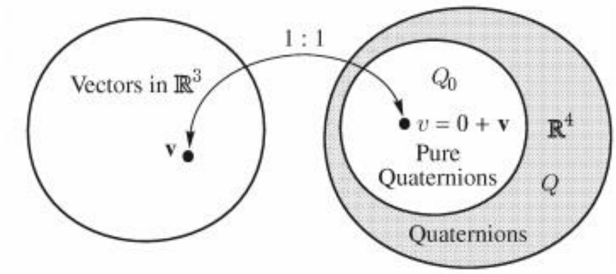
dapat juga direpresentasikan dengan quaternion.

$$\boldsymbol{q} = e^{\frac{\theta}{2}(u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}\&)}$$

$$\boldsymbol{q} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2} \qquad (17)$$

Dapat juga ditunjukkan bahwa perputaran yang diinginkan dapat digunakan pada vektor biasa

$$\mathbf{p} = (p_x, p_y, p_z) = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k} \qquad (18)$$




Sumber : [2]

**Gambar 3.1 vektor dalam dimensi tiga pada quaternion**

dalam dimensi tiga, dianggap sebagai quaternion dengan koordinat sama dengan nol, dengan mengevaluasi konjugasi dari **p** dengan **q** :

$$\mathbf{p'} = \mathbf{qpq^{-1}} \qquad (19)$$

menggunakan produk Hamilton, dimana $\mathbf{p'}=(p_x', p_y', p_z')$ adalah posisi vektor baru setelah rotasi.

## B. *Keunggulan Perputaran Quaternion*

Perhitungan perputaran menggunakan quaternion terbukti lebih unggul dibandingkan dengan menggunakan matriks perputaran. Hal ini disebabkan oleh beberapa faktor.

1. *Representasi lebih padat*

Representasi perputaran dengan quaternion (menggunakan 4 angka) lebih padat dari pada representasi perputaran dengan matriks perputaran. Matriks perputaran pada dasarnya adalah matriks ortogonal yang membutuhkan 9 angka pada dimensi tiga.

2. *Perputaran lebih halus*

Dalam aplikasi yang mementingkan grafik, perputaran yang halus sangat diperlukan. Perputaran yang halus ini berarti setiap adegan perputaran harus dapat berlangsung secara bertahap / perlahan dan tidak dalam satu langkah diselesaikan. Hal ini dapat diselesaikan dengan menggunakan kurva seperti pada *spherical linear interpolation* yang terdapat di quaternion [4]. Kasus ini lebih sulit diatasi jika menggunakan representasi perputaran yang lainnya.

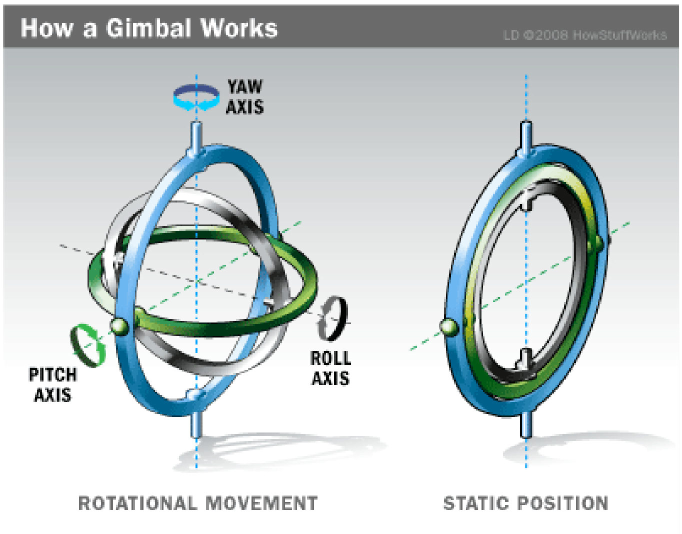
3. *Mengurangi kesalahan dan peningkatan performa*

Ketika digunakan untuk melakukan beberapa perputaran pada komputer, kesalahan karena pembulatan semakin lama akan semakin bertambah. Quaternion yang sedikit tidak akurat tetap dapat merepresentasikan perputaran meskipun setelah dinormalisasi, tetapi matriks yang sedikit tidak akurat setelah dinormalisasi mungkin menjadi matriks yang tidak ortogonal dan akan lebih sulit untuk dikembalikan untuk menjadi matriks ortogonal.

4. *Menghindari gimbal lock*

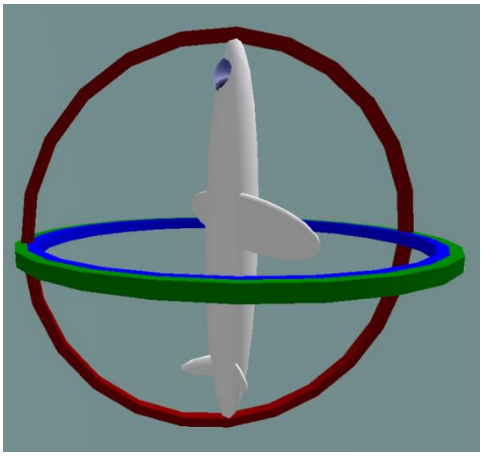
Penggunaan quaternion dapat menghindari fenomena yang disebut sebagai *gimbal lock* (kunci gimbal). *Gimbal* adalah sokongan berporos yang membuat rotasi dari objek dapat terjadi pada suatu sumbu. Sedangkan *gimbal lock* adalah fenomena kehilangan satu derajat kebebasan dalam timensi tiga. *Gimbal lock* terjadi ketika dua atau lebih sumbu *gimbal* menjadi dalam bentuk paralel, sehingga sistem akan terkunci dan merosot kedalam gerakan dua dimensi.

Salah satu fenomena *gimbal lock* yang sangat fatal di dunia nyata adalah saat penerbangan. Pesawat yang mengalami *gimbal lock* akan terkunci pada salah satu sumbu dan dapat mengakibatkan bencana fatal jika dalam menukik tajam atau naik.



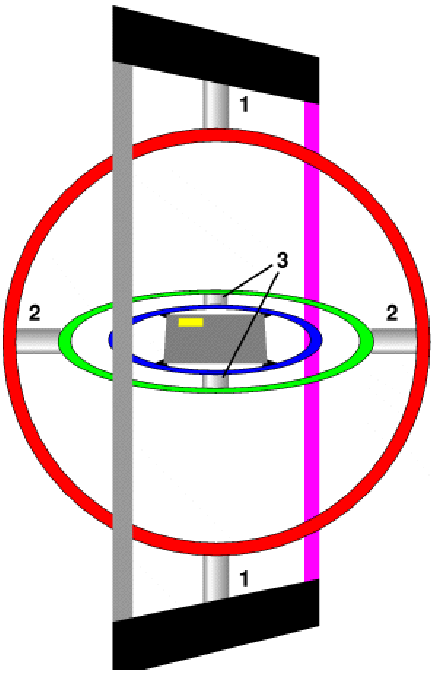

Sumber : https://coolcodea.files.wordpress.com/2013/12/gimbal-2.gif

**Gambar 3.2 Gimbal lock terjadi saat static position**



Sumber : MathPoetry (Wikipedia user)

**Gambar 3.3 Dalam keadaan gimbal lock, pesawat hanya dapat bergerak dua dimensi**




Sumber : http://history.nasa.gov/ap08fj/pics/gimbal2.gif

**Gambar 3.4 Gimbal lock dengan 2 gimbal yang paralel.**

*C. Konversi Matriks Perputaran dengan Quaternion*

*1. Quaternion ke Matriks Ortogonal*

Perkalian quaternion dan perkalian matriks ortogonal dapat digunakan untuk merepresentasikan perputaran. Jika suatu quaternion direpresentasikan sebagai qw + **i** qx + **j** qy + **k** qz, maka matriks yang ekivalen dengan itu adalah

$$\begin{bmatrix} 1-2*qy^2-2*qz^2 & 2*qx*qy-2*qz*qw & 2*qx*qz+2*qy*qw \\ 2*qx*qx+2*qz*qw & 1-2*qx^2-2*qz^2 & 2*qy*qz-2*qx*qw \\ 2*qx*qz-2*qy*qw & 2*qy*qz+2*qx*qw & 1-2*qx^2-2*qy^2 \end{bmatrix} \quad (19)$$

*Metode alternatif*

Selain menggunakan matriks 3x3 tersebut, dapat juga direpresentasika sebagai matriks 4x4

$$\begin{bmatrix} q.w & q.z & -q.y & q.x \\ -q.z & q.w & q.x & q.y \\ q.y & -q.x & q.w & q.z \\ -q.x & -q.y & -q.z & q.w \end{bmatrix}\begin{bmatrix} q.w & q.z & -q.y & q.x \\ -q.z & q.w & q.x & q.y \\ q.y & -q.x & q.w & q.z \\ -q.x & -q.y & -q.z & q.w \end{bmatrix} \quad (20)$$

*2. Matriks Ortogonal ke Quaternion*

Mengubah matriks ortogonal menjadi quaternion lebih baik dilakukan tidak secara langsung dari matriks 3x3. Hal ini disebabkan oleh kemungkinan matriks perputaran yang tidak murni perputaran karena kesalahan pembulatan. Karena itu lebih baik diubah menjadi matriks 4x4 terlebih dahulu. Misalkan Q adalah matriks rotasi, maka matriks 4x4 :

$$\frac{1}{3}\begin{bmatrix} Q_{xx}-Q_{yy}-Q_{zz} & Q_{yx}+Q_{xy} & Q_{zx}+Q_{xz} & Q_{yz}-Q_{zy} \\ Q_{yx}+Q_{xy} & Q_{yy}-Q_{xx}-Q_{zz} & Q_{zy}-Q_{yz} & Q_{zx}+Q_{xz} \\ Q_{zx}+Q_{xz} & Q_{zy}-Q_{yz} & Q_{zz}-Q_{xx}-Q_{yy} & Q_{xy}+Q_{yx} \\ Q_{yz}-Q_{zy} & Q_{zx}+Q_{xz} & Q_{xy}+Q_{yx} & Q_{xx}-Q_{yy}-Q_{zz} \end{bmatrix}$$

*D. Perbandingan antara Matriks Perputaran dan Quaternion*

| | Matriks | Quaternion |
|---|---|---|
| Jumlah data | 9 | 4 |
| Jumlah perhitungan pada rotasi beruntun | 27 perkalian, 18 penjumlahan | 16 perkalian, 12 penjumlahan |
| Jumlah berhitungan pada perputaran 3-D | 9 perkalian, 6 penjumlahan | 18 perkalian, 12 penjumlahan |

Bersadarkan data tersebut, maka quaternion jelas lebih unggul dari pada matriks perputaran. Keunggulan tersebut dapat dilihat pada bidang

- Navigasi penerbangan, penggunaan memori dikurangi sampai 55%.
- Keunggulan yang sangat tinggi untuk animasi 3-D karena detail perhitungannya.

## IV. KESIMPULAN

Perputaran dapat direpresentasikan dengan berbagai cara. Diantaranya adalah dengan matriks perputaran dan juga quaternion. Representasi quaternion diketahui sebagai representasi yang paling baik digunakan, sedangkan dengan matriks perputaran masih banyak kekurangan yang sulit untuk diatasi.

Kekuarangan matriks perputaran adalah

- Membutuhkan penyimpanan banyak untuk data
- Perputaran tidak halus
- Sulit dikoreksi setelah terjadi kesalahan pembulatan
- Dapat terjadi *gimbal lock*

*Gimbal lock* adalah suatu keadaan dimana dua atau lebih sumbu *gimbal* menjadi paralel sehingga kehilangan kemampuannya untuk dapat berputar dengan sumbu tersebut.

Salah satu cara yang paling efektif untuk menghindarinya adalah dengan menggunakan representasi quaternion. Pada representasi quaternion fenomena *gimbal lock* dapat dihindari. Selain itu, quaternion juga menghasilkan perputaran yang halus dengan *spherical linear interpolation* dan juga lebih mudah dikoreksi dan membutuhkan penyimpanan lebih sedikit.

Quaternion dan matriks perputaran dapat saling diubah menjadi bentuk yang lainnya. Lebih mudah mengubah dari matriks perputaran menjadi quaternion melalui perubahan ke matriks 4x4 terlebih dahulu.

Beberapa bidang yang terbantu dengan keunggulan representasi quaternion dalam perputaran adalah navigasi penerbangan karena penggunaan penyimpanan lebih sedikit dan juga animasi 3D karena perputarannya yang halus.

## V. REFERENSI


[1] Weisstein, Eric W. "Rotation Matrix." From MathWorld -- A Wolfram Web Resource. <http://mathworld.wolfram.com/RotationMatrix.html>
Tanggal akses : 15 Desember 2015 pukul 09.13

[2] Kuipers, Jack B. Quaternions and Rotation Sequences <http://www.emis.de/proceedings/Varna/vol1/GEOM09.pdf>
Tanggal akses : 15 Desember 2015 pukul 10.20

[3] Kuipers Jack B., Quaternions & Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University, Princeton, New Jersey 1999.

[4] Dam, Erik B., Martin Koch, dan Martin Lillholm. Quaternion, Interpolation and Animation. <http://web.mit.edu/2.998/www/QuaternionReport1.pdf>
Tanggal akses : 15 Desember 2015 pukul 12.30

[5] Baker, Martin John. Maths – Conversion Quaternion to Matrix. <http://www.euclideanspace.com/maths/geometry/rotations/conversions/quaternionToMatrix/>
Tanggal akses : 15 Desember 2015 pukul 13.10

[6] Kang, Sungwoo. Quaernion and its Application. <http://profstewart.org/pm1/talks/20100157_Quaternion.pdf>
Tanggal akses : 15 Desember 2015 pukul 13.30

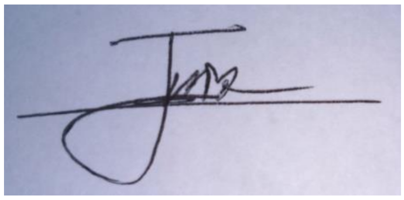
[7] Salamin, Eugene. Application of Quaternions to Computation with Rotations. <http://people.csail.mit.edu/bkph/articles/Stanford_AI_WP79-Salamin.pdf>
Tanggal akses : 15 Desember 2015 pukul 13.45

[8] Vince, John. Geometric Algebra for Computer Graphic. <https://books.google.co.id/books?id=32deMZQ6TG8C&printsec=frontcover&hl=id&source=gbs_atb#v=onepage&q&f=false>
Tanggal akses : 15 Desember 2015 pukul 14.07


## 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, 15 Desember 2015



Jeremia Jason Lasiman, 13514021