

# Penerapan Sistem Persamaan Linear pada Sistem *Shot Meter* Game NBA 2K

Ardell Aghna Mahendra dan 13523151<sup>1,2</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[13523151@std.stei.itb.ac.id](mailto:13523151@std.stei.itb.ac.id), [ddaghna@gmail.com](mailto:ddaghna@gmail.com)

**Abstrak**—Sistem *shot meter* dalam game NBA 2K merupakan mekanisme yang penting dalam menentukan keberhasilan tembakan pemain untuk mencetak poin. Makalah ini bertujuan untuk mengeksplorasi penerapan sistem persamaan linear dalam mengimplementasikan sistem *shot meter* yang realistis. Dengan mempertimbangkan empat faktor utama, yaitu jarak tembakan dari ring, statistik tembakan pemain, tingkat kelelahan pemain, dan tekanan dari pertahanan lawan, probabilitas keberhasilan tembakan dihitung menggunakan sistem persamaan linear beserta *timing window* dan *release zone*. Hasil yang didapatkan dari perhitungan probabilitas menunjukkan bahwa sistem persamaan linear berhasil menampilkan akurasi tembakan yang mendekati statistik aktual NBA dengan selisih 11,3 – 13,5%, serta *timing window* dan *release zone* yang menyesuaikan dengan kondisi pemain dan permainan memberikan pengalaman bermain yang lebih nyata dan permainan yang lebih realistis.

**Keywords**—Game NBA 2K, Probabilitas Tembakan, Sistem Persamaan Linear, *Shot Meter*.

## I. PENDAHULUAN

Perkembangan teknologi *game* dalam dunia ini semakin maju, memberikan pengalaman yang semakin realistis kepada para pemainnya. Salah satu aspek penting dalam *game* olahraga seperti *game* basket, yaitu NBA 2K adalah sistem yang mengatur keberhasilan tembakan pemain atau yang sering disebut *shot meter*. Sistem ini sangat penting karena menentukan jumlah poin yang akan dicetak oleh pemain sekaligus menentukan tingkat kesulitan dan realisme permainan.

Berbagai franchise *game* terus berupaya untuk meningkatkan pengalaman bermain bagi para penggemarnya termasuk *game* NBA 2K. Salah satu tantangannya utamanya adalah menciptakan sistem *shot meter* yang dapat mempresentasikan kemampuan pemain basket profesional secara akurat. Sistem *shot meter* yang terlalu sulit atau mudah akan mempengaruhi pengalaman bermain para penggemar dan realisme *game* itu sendiri. Penerapan matematika, khususnya sistem persamaan linear

menawarkan solusi untuk mengimplementasikan *shot meter* yang lebih realistis. Dengan mempertimbangkan berbagai variabel seperti jarak tembakan dari ring, statistik pemain, tingkat kelelahan pemain, dan tekanan dari pertahanan lawan, sistem persamaan linear dapat menghasilkan probabilitas keberhasilan tembakan yang lebih akurat mendekati statistik pemain basket profesional.

Oleh karena itu, makalah ini bertujuan untuk mengeksplorasi penerapan sistem persamaan linear dalam mengimplementasikan sistem *shot meter* yang realistis dengan mengembangkan model matematis yang mempertimbangkan empat faktor utama yang mempengaruhi keberhasilan tembakan serta menghasilkan *timing window* dan *release zone* yang sesuai dengan kondisi permainan.

## II. LANDASAN TEORI

### A. Sistem Persamaan Linear

#### A.1. Definisi Sistem Persamaan Linear

Sistem Persamaan linear atau yang sering disingkat SPL adalah suatu sistem yang terdiri dari satu atau lebih persamaan linear dengan variabel-variabel yang sama. Sebuah persamaan disebut linear jika dapat dinyatakan dalam bentuk  $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$  dengan  $a_1..a_n$  adalah koefisien dan  $b$  adalah konstanta. Lalu, sistem persamaan linear dengan  $m$  persamaan dan  $n$  variabel dapat dinyatakan dalam bentuk:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

Solusi dari SPL dapat berupa solusi unik, banyak solusi, atau tidak memiliki solusi. Solusi unik adalah kondisi ketika sistem memiliki tepat satu solusi. Hal tersebut terjadi karena setelah dilakukan eliminasi, matriks augmented akhir berbentuk segitiga sampai baris terakhir.

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right]$$

**Gambar 1.** Contoh solusi unik  
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf>)

Banyak solusi adalah kondisi ketika sistem memiliki solusi yang tak hingga. Hal tersebut terjadi karena setelah dilakukan eliminasi, terdapat satu atau lebih baris matriks yang variabel dan konstantanya bernilai 0.

$$\left[ \begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

**Gambar 2.** Contoh banyak solusi  
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf>)

Tidak ada solusi adalah kondisi ketika sistem tidak memiliki solusi. Hal tersebut terjadi karena setelah dilakukan eliminasi, terdapat baris yang seluruh variabelnya bernilai 0, tetapi konstantanya tidak bernilai 0.

$$\left[ \begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

**Gambar 3.** Contoh tidak ada solusi  
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf>)

## A.2. Metode Eliminasi Gauss

Metode eliminasi Gauss adalah metode yang mengubah matriks augmented menjadi matriks eselon baris melalui operasi baris elementer (OBE). OBE atau Operasi Baris Elementer sendiri memiliki 3 jenis operasi dasar, yaitu menukar posisi dua baris, mengalikan suatu baris dengan konstanta tidak nol, dan menambahkan kelipatan suatu baris ke baris lainnya. Metode eliminasi Gauss ditemukan oleh Carl Friedrich Gauss dan metode ini melibatkan dua tahap utama, yaitu tahap eliminasi maju, yaitu mengubah matriks augmented menjadi matriks eselon baris dan tahap substitusi mundur, yaitu menentukan nilai variabel dari baris terakhir ke baris pertama. Metode eliminasi Gauss dapat digunakan dalam penyelesaian sistem persamaan linear karena kesederhanaannya, efisiensi dalam kalkulasi real-timed, dan mampu menangani sistem dengan banyak

variabel dan perubahan nilai variabel dengan cepat jika diimplementasikan dalam sebuah program.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{bmatrix} \sim \text{OBE} \sim \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & * \end{bmatrix}$$

**Gambar 4.** Metode eliminasi Gauss  
(Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf>)

## B. Teori Probabilitas Tembakan

Probabilitas dalam konteks *game* adalah ukuran numerik yang menunjukkan kemungkinan terjadinya suatu kejadian dalam ruang sampel yang terdefinisi. Nilai probabilitas akan selalu berada di antara 0 dan 1. Dalam konteks *shot meter* NBA 2K, ruang sampel terdiri dari dua kemungkinan hasil, yaitu berhasil atau gagal. Probabilitas keberhasilan tembakan dipengaruhi oleh berbagai faktor yaitu jarak tembakan dari ring, *rating* atau statistik tembakan pemain, kelelahan pemain, dan intensitas/tekanan pertahanan lawan. Probabilitas keberhasilan tersebut dapat dimodelkan menggunakan fungsi probabilitas berikut:

$$F(X) = P(j, s, f, p)$$

yang mana X adalah kejadian tembakan berhasil, j adalah jarak tembakan dari ring (dalam feet), s adalah *rating* atau statistik tembakan pemain, f adalah kelelahan pemain, dan p adalah intensitas/tekanan pertahanan lawan.

## C. NBA 2K

NBA 2K adalah seri video *game* simulasi bola basket yang dikembangkan oleh Visual Concepts dan diterbitkan oleh 2K Sports. Game ini pertama kali rilis pada tahun 1999 dengan nama NBA 2K dan terus berkembang menjadi salah satu franchise *game* olahraga terpopuler dengan perlisian tiap tahun. NBA 2K secara resmi juga berkolaborasi dengan NBA atau National Basketball Association untuk memberikan pengalaman basket yang lebih nyata.



**Gambar 5.** NBA 2K berkolaborasi dengan NBA untuk kompetisi Summer league  
(Sumber:

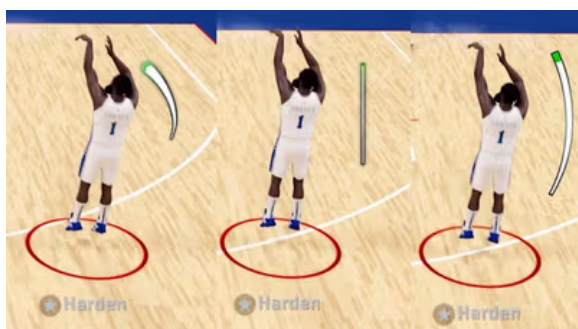
<https://www.raptorshq.com/2024/7/22/24203641/summer-league-toronto-raptors-vs-san-antonio-spurs-89-100-raps-fall-in-final-game>

NBA 2K tidak hanya menawarkan mode permainan biasa, tetapi terdapat banyak mode lainnya seperti MyCareer, MyTeam, MyLeague, dan TheCity. Dengan banyaknya mode yang ditawarkan, para penggemar dapat memainkan sesuai dengan keinginan mereka, seperti jika ingin main sebagai pemain NBA ia bisa bermain MyCareer atau jika ingin menjadi manajemen sebuah tim NBA ia bisa bermain MyTeam atau My League. NBA 2K berusaha mensimulasikan pengalaman bermain bola basket profesional dengan menggabungkan berbagai elemen seperti grafis yang realistis, mekanisme permainan yang kompleks, serta statistik pemain yang akurat berdasarkan performa mereka di NBA. Statistik tersebut nantinya akan berpengaruh kepada *rating* pemain di dalam *game*. Setiap pemain akan memiliki nilai numerik (0-99) untuk berbagai atribut, seperti *Three Point Shot*, *Mid Range Shot*, *Layup*, *Ball Handle*, dan *Defense*. *Rating* para pemain akan diperbarui secara berkala berdasarkan perubahan statistik para pemain dan nantinya akan berpengaruh kepada performa mereka di dalam *game*.

#### D. Shot Meter

*Shot meter* merupakan mekanisme visual dalam *game* NBA 2K yang berfungsi untuk mempresentasikan, mengatur, dan menentukan keberhasilan tembakan pemain. Sistem ini pertama kali diperkenalkan dalam NBA 2K15 sebagai solusi untuk memberikan pengalaman bermain yang lebih realistis dan terukur pada para penggemar basket sekaligus *game* NBA 2K. *Shot meter* menggabungkan elemen visual dan matematis untuk menentukan hasil tembakan berdasarkan berbagai parameter.

Komponen utama *shot meter* terdiri dari *bar timing* yang bergerak dan *release zone* yang menandakan area optimal untuk melepaskan tembakan. Ukuran *release zone* bervariasi dan dihitung menggunakan sistem persamaan linear yang mempertimbangkan statistik pemain dan faktor situasional. Apabila seorang pemain berhasil tepat menaruh *bar timing* ke dalam *release zone* dia akan dikategorikan sebagai *green* dan tembakan dia akan otomatis masuk, sedangkan apabila *bar timing* berada di luar *release zone*, maka tembakan kemungkinan tembakan itu masuk mengecil.



**Gambar 6.** Contoh *shot meter* yang ada di dalam *game* NBA 2K

(Sumber: <https://realsport101.com/article/how-to-change-your-nba-2k23-shot-meter>)



**Gambar 7.** Visualisasi *release zone* dan *bar timing* dengan lebih jelas

(Sumber: <https://www.vgr.com/nba-2k21-shooting-with-new-shot-meter/>)

Untuk menciptakan simulasi yang akurat, sistem *shot meter* menggunakan data statistik NBA sebagai referensi utama. *Field Goal Percentage* (FG%) merupakan statistik dasar yang menunjukkan rasio antara tembakan yang berhasil dengan total percobaan tembakan. *Three Point Percentage* (3P%) secara khusus mengukur keberhasilan tembakan tiga angka, sementara itu *Contested Shot Percentage* menunjukkan kemampuan pemain mencetak poin di bawah tekanan pertahanan pemain lawan.

Pada *shot meter*, statistik NBA ini diintegrasikan ke dalam sistem dengan serangkaian persamaan linear yang menghasilkan probabilitas dasar untuk setiap jenis tembakan. Sebagai contoh, pemain seperti Stephen Curry yang memiliki persentase tembakan tiga angka tinggi akan mendapatkan probabilitas dasar yang lebih tinggi untuk tembakan tiga angka. Probabilitas ini kemudian disesuaikan secara real-time berdasarkan faktor situasional seperti jarak tembakan dan tekanan pertahanan. Nantinya sistem persamaan linear untuk sistem *shot meter* NBA 2K akan memiliki persamaan  $a_1j + a_2s + a_3f + a_4p = P$  yang mana  $a$  adalah bobot untuk masing-masing faktor,  $j$  adalah jarak pemain dari ring,  $s$  adalah statistik,  $f$  adalah kelelahan pemain,  $p$  adalah tekanan dari pertahanan lawan, dan  $P$  adalah probabilitas keberhasilan tembakan dari data historis.

### III. IMPLEMENTASI

#### A. Persiapan Dataset

Data yang digunakan pada makalah ini akan menggunakan statistik NBA 2K25 [6] dan data simulasi untuk mengimplementasikan sistem *shot meter*. Pemain yang akan digunakan dalam implementasi ini, yaitu LeBron James, Stephen Curry, dan James Harden. Data yang digunakan mencakup empat parameter/variabel utama yang mempengaruhi probabilitas tembakan.

**Tabel 1.** Dataset yang akan digunakan dalam implementasi

Parameter/variabel	Lebron James	Stephen Curry	James Harden
Jarak Pemain dari Ring (feet)	25	25	25
Three Point Rating	79	99	80
Mid Range Shooting Rating	74	96	88
Close Shot Rating	93	88	73
Tingkat Kelelahan (%)	20, 50, 80	20, 50, 80	20, 50, 80
Tekanan Pertahanan Lawan (%)	25, 50, 75	25, 50, 75	25, 50, 75

## B. Pengolahan Data

Setelah mendapatkan data yang diperlukan, data-data tersebut perlu dinormalisasi terlebih dahulu sebelum diterapkan dalam sistem persamaan linear agar berada di rentang yang seragam. Proses normalisasi dilakukan dengan cara, yaitu untuk jarak pemain dari ring dinormalisasi dengan cara jarak dibagi dengan 25, untuk *rating three point*, *mid range shooting* dan *close shot*, nilai *rating* dibagi dengan 99, untuk tingkat kelelahan, nilai kelelahan dibagi dengan 100, dan untuk tekanan pertahanan lawan, nilai tekanan dibagi dengan 100.

```
self.data = np.array([
    # LeBron James
    [0.20, 0.94, 0.20, 0.25, 0.85],
    [0.60, 0.75, 0.50, 0.50, 0.65],
    [1.00, 0.80, 0.80, 0.75, 0.45],

    # Stephen Curry
    [0.20, 0.89, 0.20, 0.25, 0.90],
    [0.60, 0.97, 0.50, 0.50, 0.75],
    [1.00, 1.00, 0.80, 0.75, 0.60],

    # James Harden
    [0.20, 0.74, 0.20, 0.25, 0.87],
    [0.60, 0.89, 0.50, 0.50, 0.57],
    [1.00, 0.81, 0.80, 0.75, 0.47],
])
self.coefficients = None
```

Gambar 8. Data yang akan dipakai (Sumber: Dokumentasi Pribadi)

Implementasi sistem persamaan linear kemudian dilakukan dengan menggunakan python dengan library numpy untuk operasi matriks. Data pada gambar 8 disusun dalam bentuk matriks augmented yang mempresentasikan sistem persamaan linear  $a_{1j} + a_{2s} + a_{3f} + a_{4p} = P_i$ . Selanjutnya penyelesaian sistem akan menggunakan metode least squares untuk mendapatkan koefisien optimal karena terdapat lebih banyak data historis (persamaan) dibandingkan jumlah koefisien yang dicari. Metode least squares akan mencari nilai koefisien yang meminimalkan jumlah kuadrat *error* antara prediksi model dan data aktual.

```
def hitung_koefisien(self):
    print("a_j + a_2s + a_3f + a_4p = P_i")
    print("\nEquations:")

    A = self.data[:, :4]
    b = self.data[:, 4]
    for i, row in enumerate(self.data):
        print(f"({row[0]:.3f})a_1 + ({row[1]:.3f})a_2 + ({row[2]:.3f})a_3 + ({row[3]:.3f})a_4 = ({row[4]:.3f}")
    self.coefficients = np.linalg.lstsq(A, b, rcond=None)[0]

    print("\nCoefficients:")
    print("a_1 (jarak) = (self.coefficients[0]:.3f)")
    print("a_2 (rating) = (self.coefficients[1]:.3f)")
    print("a_3 (fatigue) = (self.coefficients[2]:.3f)")
    print("a_4 (defense) = (self.coefficients[3]:.3f)")
    return self.coefficients
```

Gambar 9. Program untuk mendapatkan koefisien optimal (Sumber: Dokumentasi Pribadi)

## C. Implementasi Shot Meter

*Shot meter* diimplementasikan dalam kelas *shot meter* yang menggunakan hasil koefisien dari class *SPLSolver*.

```
class ShotMeter:
    def __init__(self, coefficients):
        self.coefficients = coefficients
        self.base_window = 150 # Base timing window dalam ms
```

Gambar 10. Class *shot meter* (Sumber: Dokumentasi Pribadi)

Kemudian dilakukan perhitungan probabilitas. *timing window*, dan *release zone* dengan mempertimbangkan seluruh faktor.

```
def hitung_matriks(self, distance, rating, fatigue, defense):
    # Normalisasi input
    jarak = distance / 25.0
    kecapean = fatigue / 100.0
    pertahanan = defense / 100.0

    params = np.array([
        jarak,
        rating,
        (1 - kecapean),
        (1 - pertahanan)
    ])

    # raw probability
    raw_prob = np.dot(params, self.coefficients)

    # Normalisasi ke range 0-1
    probability = max(0.1, min(0.95, raw_prob))

    base_timing = self.base_window * probability
    fatigue_factor = 1 - (kecapean * 0.3)
    defense_factor = 1 - (pertahanan * 0.3)
    timing_ms = base_timing * fatigue_factor * defense_factor
    release_zone_ms = timing_ms * 0.2

    return probability, timing_ms, release_zone_ms
```

Gambar 11. Program untuk menghitung probabilitas menggunakan SPL, *timing window*, dan *release zone* (Sumber: Dokumentasi Pribadi)

## IV. HASIL DAN PEMBAHASAN

Program yang telah dibuat akan memberikan *output* yaitu probabilitas tembakan dari masing-masing keadaan. Pemain dengan *rating* lebih tinggi pastinya akan memiliki probabilitas yang lebih besar. Program juga menampilkan *release zone* dalam bentuk *milliseconds* (ms) dan *timing window* dalam bentuk *milliseconds* (ms)

LeBron James					
Distance: 5ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 95.0%	Timing Window: 113.9ms	Release Zone: 22.8ms	
Fatigue: 20%	Defense: 75%	Success: 93.0%	Timing Window: 102.3ms	Release Zone: 20.5ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 95.0%	Timing Window: 103.0ms	Release Zone: 20.6ms	
Fatigue: 50%	Defense: 75%	Success: 88.1%	Timing Window: 87.8ms	Release Zone: 17.6ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 95.0%	Timing Window: 92.1ms	Release Zone: 18.4ms	
Fatigue: 80%	Defense: 75%	Success: 84.1%	Timing Window: 74.3ms	Release Zone: 14.9ms	
Distance: 15ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 70.3%	Timing Window: 84.2ms	Release Zone: 16.8ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 65.6%	Timing Window: 71.0ms	Release Zone: 14.2ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 60.8%	Timing Window: 58.9ms	Release Zone: 11.8ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	
Distance: 25ft					
Fatigue: 20%	Defense: 25%	Success: 47.0%	Timing Window: 61.3ms	Release Zone: 12.3ms	
Fatigue: 20%	Defense: 50%	Success: 10.0%	Timing Window: 12.0ms	Release Zone: 2.4ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 45.9%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 10.0%	Timing Window: 10.8ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 37.5%	Timing Window: 39.6ms	Release Zone: 7.9ms	
Fatigue: 80%	Defense: 50%	Success: 10.0%	Timing Window: 9.7ms	Release Zone: 1.9ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	

**Gambar 12.** Hasil dari perhitungan probabilitas, *timing window*, dan *release zone* LeBron James  
(Sumber: Dokumentasi Pribadi)

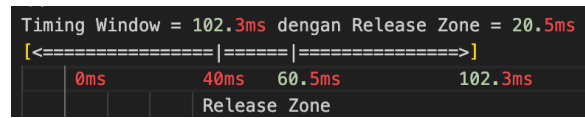
Stephen Curry					
Distance: 5ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 95.0%	Timing Window: 113.9ms	Release Zone: 22.8ms	
Fatigue: 20%	Defense: 75%	Success: 95.0%	Timing Window: 103.8ms	Release Zone: 20.8ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 95.0%	Timing Window: 103.0ms	Release Zone: 20.6ms	
Fatigue: 50%	Defense: 75%	Success: 95.0%	Timing Window: 93.9ms	Release Zone: 18.8ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 95.0%	Timing Window: 92.1ms	Release Zone: 18.4ms	
Fatigue: 80%	Defense: 75%	Success: 91.2%	Timing Window: 80.6ms	Release Zone: 16.1ms	
Distance: 15ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 77.4%	Timing Window: 92.8ms	Release Zone: 18.6ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 72.7%	Timing Window: 78.8ms	Release Zone: 15.8ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 67.9%	Timing Window: 65.8ms	Release Zone: 13.2ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	
Distance: 25ft					
Fatigue: 20%	Defense: 25%	Success: 54.1%	Timing Window: 70.6ms	Release Zone: 14.1ms	
Fatigue: 20%	Defense: 50%	Success: 10.0%	Timing Window: 12.0ms	Release Zone: 2.4ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 49.4%	Timing Window: 58.2ms	Release Zone: 11.6ms	
Fatigue: 50%	Defense: 50%	Success: 10.0%	Timing Window: 10.8ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 44.6%	Timing Window: 47.1ms	Release Zone: 9.4ms	
Fatigue: 80%	Defense: 50%	Success: 10.0%	Timing Window: 9.7ms	Release Zone: 1.9ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	
James Harden					
Distance: 5ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 95.0%	Timing Window: 113.9ms	Release Zone: 22.8ms	
Fatigue: 20%	Defense: 75%	Success: 94.0%	Timing Window: 102.7ms	Release Zone: 20.5ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 95.0%	Timing Window: 103.0ms	Release Zone: 20.6ms	
Fatigue: 50%	Defense: 75%	Success: 95.0%	Timing Window: 93.9ms	Release Zone: 18.8ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 95.0%	Timing Window: 92.1ms	Release Zone: 18.4ms	
Fatigue: 80%	Defense: 75%	Success: 84.5%	Timing Window: 74.6ms	Release Zone: 14.9ms	
Distance: 15ft					
Fatigue: 20%	Defense: 25%	Success: 95.0%	Timing Window: 123.9ms	Release Zone: 24.8ms	
Fatigue: 20%	Defense: 50%	Success: 70.6%	Timing Window: 84.7ms	Release Zone: 16.8ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 95.0%	Timing Window: 112.0ms	Release Zone: 22.4ms	
Fatigue: 50%	Defense: 50%	Success: 65.6%	Timing Window: 71.0ms	Release Zone: 14.2ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 95.0%	Timing Window: 100.2ms	Release Zone: 20.0ms	
Fatigue: 80%	Defense: 50%	Success: 61.2%	Timing Window: 59.3ms	Release Zone: 11.9ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	
Distance: 25ft					
Fatigue: 20%	Defense: 25%	Success: 47.3%	Timing Window: 61.7ms	Release Zone: 12.3ms	
Fatigue: 20%	Defense: 50%	Success: 10.0%	Timing Window: 12.0ms	Release Zone: 2.4ms	
Fatigue: 20%	Defense: 75%	Success: 10.0%	Timing Window: 10.9ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 25%	Success: 42.0%	Timing Window: 50.2ms	Release Zone: 10.8ms	
Fatigue: 50%	Defense: 50%	Success: 10.0%	Timing Window: 10.8ms	Release Zone: 2.2ms	
Fatigue: 50%	Defense: 75%	Success: 10.0%	Timing Window: 9.9ms	Release Zone: 2.0ms	
Fatigue: 80%	Defense: 25%	Success: 37.0%	Timing Window: 39.9ms	Release Zone: 8.0ms	
Fatigue: 80%	Defense: 50%	Success: 10.0%	Timing Window: 9.7ms	Release Zone: 1.9ms	
Fatigue: 80%	Defense: 75%	Success: 10.0%	Timing Window: 8.8ms	Release Zone: 1.8ms	

**Gambar 13.** Hasil dari perhitungan probabilitas, *timing window*, dan *release zone* Stephen Curry dan James Harden  
(Sumber: Dokumentasi Pribadi)

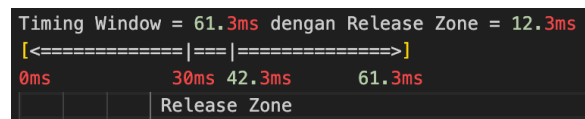
Berdasarkan hasil yang diperoleh, dapat terlihat bahwa semakin jaraknya jauh, maka probabilitas tembakan tersebut juga makin mengecil. Faktor seperti tingkat kelelahan pemain dan tekanan pertahanan juga mempengaruhi probabilitas tembakan yang mana terlihat apabila pemain lebih capek dan pertahanan makin ketat, probabilitas tembakan juga makin mengecil. Lalu, statistik

atau *rating* tembakan pemain juga mempengaruhi probabilitas tembakan, sebagai contoh Stephen Curry dengan *rating* tembakan tiga angka (25 feet) paling besar memiliki probabilitas yang paling besar dibandingkan dengan LeBron James atau James Harden.

*Timing Window* dan *release zone* juga terpengaruh akibat dari perbedaan jarak, statistik atau *rating shooting* pemain, tingkat kelelahan pemain, dan tekanan pertahanan lawan. *Timing Window* sendiri adalah rentang waktu total yang diberikan untuk pemain menekan tombol tembak dalam *game* NBA 2K. Dapat dilihat pada hasil yang didapatkan, semakin jauh jarak pemain dari ring atau *rating* tembakannya makin rendah atau tingkat kelelahannya makin tinggi atau tekanan pertahanan lawan makin tinggi, maka *timing window* (ms) dan *release zone* (ms) akan makin mengecil. Dengan mengecilnya *timing window* (ms) dan *release zone* (ms), maka tingkat kesulitan pemain untuk mendapatkan *perfect timing release* atau *green* akan makin sulit. Hal ini berhasil memberikan pengalaman yang realistis di mana pemain NBA 2K tidak akan selalu gampang untuk memasukkan bola ke dalam ring.



**Gambar 14.** Contoh visualisasi dari *shot meter* dengan *release zone* 20.5 ms  
(Sumber: Dokumentasi Pribadi)



**Gambar 15.** Contoh visualisasi dari *shot meter* dengan *release zone* 12.3 ms  
(Sumber: Dokumentasi Pribadi)

Hasil probabilitas yang didapatkan dapat dibandingkan dengan data aktual NBA [7]. Seperti contohnya apabila dibandingkan antara probabilitas tembakan tiga angka (25 feet) dengan kondisi paling tidak capek dan tekanan pertahanan lawan paling rendah dengan data aktual NBA, didapatkan bahwa selisih antara probabilitas yang diperoleh dengan data aktual NBA, yaitu 11,3 – 13,5%. Probabilitas yang diperoleh memiliki persentase yang lebih besar dikarenakan pada *game* NBA 2K terdapat faktor lain, yaitu skill pemain dalam mengontrol *timing release* serta *game* dirancang lebih *forgiving* untuk pengalaman yang lebih baik kepada para pengguna.

**Tabel 2.** Perbandingan hasil probabilitas dengan data aktual NBA

Pemain	Hasil Probabilitas	Data Aktual NBA
Stephen Curry	54,1%	40,6%
Lebron James	47%	35,7%
James Harden	47,3%	34,1%

## V. KESIMPULAN

Bekasi, 30 Desember 2024

Penerapan sistem persamaan linear terbukti dapat digunakan dalam membuat sistem *shot meter* NBA 2K. Dengan variabel yang berisikan faktor yang mempengaruhi tembakan pemain menghasilkan perhitungan probabilitas tembakan yang tidak terlalu jauh dengan data asli pemain NBA tersebut. Lalu, dengan *timing window* dan *release zone* yang menyesuaikan dengan kondisi pemain dan permainan memberikan pengalaman bermain yang lebih nyata dan permainan yang lebih realistis, yang mana penggemar dapat merasakan tantangan *shooting* di berbagai kondisi layaknya melakukan *shooting* di pertandingan asli NBA.



Ardell Aghna Mahendra 13523151

## VI. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan syukur kepada Allah Swt karena dengan rahmatnya, makalah dengan judul “Penerapan Sistem Persamaan Linear pada Sistem *Shot Meter Game* NBA 2K” ini dapat terselesaikan dengan baik. Penulis juga ingin menyampaikan terima kasih kepada pak Dr. Judhi Santoso, M.Sc. dan pak Arrival Dwi Sentosa, S.Kom., M.T. selaku dosen mata kuliah Aljabar Linier dan Geometri IF2123 yang telah mengajar dan membimbing penulis selama satu semester ini. Terima kasih juga kepada orang tua penulis yang selalu memberikan semangat agar penulis dapat menyelesaikan makalah ini, serta pihak-pihak lain yang tidak dapat disebutkan satu-satu.

## REFERENSI

- [1] Munir, Rinaldi. (2023). Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss). <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf>, diakses pada 28 Desember 2024.
- [2] Munir, Rinaldi. (2023). Sistem persamaan linier (Bagian 2: Tiga kemungkinan solusi sistem persamaan linier). <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf>, diakses pada 28 Desember 2024.
- [3] Cuemath. (2024). Teori Probabilitas. <https://www.cuemath.com/data/probability-theory/>, diakses pada 28 Desember 2024.
- [4] Sortir A Paris. (2024). NBA 2K24: Pendapat Kami Tentang Permainan Bola Basket 2K. <https://www.sortiraparis.com/id/hobi/bermain-game/articles/301892-nba-2k24-pendapat-kami-tentang-permainan-bola-basket-2k>, diakses pada 28 Desember 2024.
- [5] Gunawan, Yufianto. (2014). Review NBA 2K15: Paduan Visual dan Gameplay Keren. <https://jagatplay.com/2014/10/pc-2/review-nba-2k15-paduan-visual-dan-gameplay-keren/2/>, diakses pada 28 Desember 2024.
- [6] 2KRatings. (2024). NBA 2K Ratings. <https://www.2kratings.com>, diakses pada 29 Desember 2024.
- [7] Basketball Reference. (2024). NBA Shooting Stats 2025. [https://www.basketball-reference.com/leagues/NBA\\_2025\\_shooting.html](https://www.basketball-reference.com/leagues/NBA_2025_shooting.html), diakses pada 30 Desember 2024.

## 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.