

Penerapan *Singular Value Decomposition* (SVD) untuk Analisis Kontribusi Pemain dalam Olahraga Basket

Raka Daffa Iftikhaar - 13523018¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
raka.daffa2005@gmail.com, 13523018@std.stei.itb.ac.id

Abstrak - Analisis kontribusi pemain dalam sebuah liga kompetitif menjadi sebuah aspek penting dalam mengevaluasi performa pemain dalam sebuah tim dan menentukan strategi yang cocok untuk tim tersebut. *Singular Value Decomposition* (SVD) sebagai teknik analisis matriks menjadi salah satu metode yang bisa digunakan untuk mengukur performa kontribusi seorang pemain. Penelitian ini bertujuan untuk menerapkan SVD dalam pengolahan data statistik pemain. Hasil penelitian menunjukkan bahwa SVD mampu menjadi sebuah alat analisis kontribusi seorang pemain dalam sebuah tim. Temuan ini memberikan panduan praktis bagi pelatih dan analisis olahraga dalam menyusun strategi dan menilai efektivitas seorang pemain dalam sebuah tim

Kata Kunci - *Singular Value Decomposition*, SVD, Kontribusi Pemain, Analisis Basket.

I. PENDAHULUAN

NBA atau *National Basketball Association* merupakan salah satu kompetisi olahraga basket profesional yang sangat populer. NBA sudah dikenal orang-orang semenjak tahun 1950-an dan sekarang sudah menjadi simbol olahraga basket dunia. Liga ini menjadi wadah bagi pemain-pemain terbaik di dunia. Sebuah kompetisi tidak akan lepas dari penggemar. NBA menjadi salah satu liga dengan penggemar terbanyak karena NBA menyajikan pertandingan basket yang menarik dan kompetitif. Sebagai penggemar, pastinya menginginkan tim yang didukungnya mempunyai performa terbaik, baik dari segi manajemen tim ataupun manajemen pemain. Sudah banyak analisis-analisis yang berdasar pada performa sebuah tim di sosial media, namun terkadang sebagai penggemar rasanya masih terdapat hal yang kurang dari analisis yang diberikan oleh pihak lain tersebut dan memilih untuk melakukan analisis secara mandiri.

Proses analisis sebuah tim dimulai dari kontribusi tiap pemainnya terlebih dahulu. Oleh karena itu, pembuatan penelitian tentang analisis persentase kontribusi pemain pada sebuah tim menjadi hal yang penting untuk dilakukan terutama untuk penggemar NBA. Analisis yang dilakukan disini menggunakan pendekatan SVD atau *Singular Value Decomposition* matriks. Pemilihan SVD sebagai pendekatan karena SVD sering digunakan untuk analisis data yang kompleks dan berbasis matriks. Pada penelitian ini, data yang digunakan adalah data tim

Washington Wizards musim 2024/2025 dengan pembatasan sebelum bulan Desember 2024. Hal tersebut dilakukan karena tim Washington Wizards merupakan tim dengan rekor terburuk per bulan Desember 2024 dan pemilihan data sebelum bulan Desember 2024 karena musim ini masih berjalan dan data akan menjadi lebih realistis terhadap performa para pemain musim ini. Walaupun demikian, diharapkan penelitian ini bisa digunakan lagi kedepannya dengan tim ataupun data yang berbeda. Selain itu juga diharapkan penelitian ini bisa digunakan untuk memberikan contoh penggunaan metode SVD dalam analisis olahraga.

II. METODE PENELITIAN

Penelitian dilakukan dengan metode kuantitatif dimana penulis menggunakan data yang berisi statistik pemain NBA dengan melakukan pengerucutan kepada tim Washington Wizards musim 2024/2025 sebelum bulan Desember 2024. Data statistik yang digunakan berupa sepuluh komponen per pertandingan yaitu *points*, *rebounds*, *assists*, *steals*, *blocks*, *turnovers*, persentase tembakan berhasil masuk, persentase tembakan tiga poin yang berhasil masuk, persentase tembakan bebas yang berhasil masuk, dan hasil (+/-). Sepuluh komponen tersebut diambil dari *traditional splits* yang menjadi dasar analisis utama dari performa seorang pemain secara individu ataupun kontribusinya terhadap tim. Selain dari sepuluh komponen tersebut, data yang digunakan juga penulis saring terkhusus untuk pemain yang memiliki menit per pertandingan di atas rata-rata. Hal itu ditujukan agar penelitian ini lebih berfokus pada peran pemain-pemain dalam rotasi utama permainan dan menganalisis apakah rotasi yang telah dilakukan efektif atau tidak.

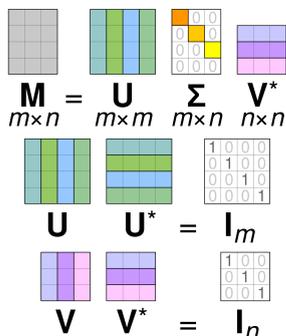
Data diambil dari situs resmi NBA yang sudah disaring dan sesuai dengan kebutuhan penelitian ini. Data disimpan dalam file csv. Data yang sudah disimpan kemudian disusun dalam sebuah matriks. Komponen baris untuk pemain dan komponen kolom untuk statistik yang penulis sebutkan di atas. Setelah itu data diproses menggunakan metode SVD yang menghasilkan persentase kontribusi pemain dalam tim dan dianalisis untuk mendapatkan kesimpulan. penulis menggunakan program python untuk dekomposisi SVD agar lebih

mudah untuk mendapatkan output dan memberikan kesimpulan akhir pada penelitian kali ini.

III. DASAR TEORI

A. Singular Value Decomposition (SVD)

Metode SVD adalah suatu metode yang mendekomposisikan matriks suatu matriks menjadi tiga komponen matriks lainnya. Suatu matriks M dengan ukuran $m \times n$ didekomposisi menjadi $U\Sigma V^T$. Matriks U merupakan matriks orthogonal berukuran $m \times m$, matriks Σ merupakan matriks diagonal yang berukuran $m \times n$ dengan nilai riil tidak negatif yang disebut nilai singular, dan matriks V adalah matriks uniter berukuran $n \times n$. Bisa dijelaskan juga bahwa matriks Σ merupakan matriks diagonal dengan elemen diagonalnya berupa nilai-nilai singular dari matriks M , sedangkan matriks U dan V merupakan matriks-matriks yang kolom-kolomnya berupa vektor singular kiri dan vektor singular kanan dari matriks M untuk nilai singular yang bersesuaian (Ariyanti 35).



Gambar 1. Dekomposisi Matriks

Sumber: <https://yeong-jin-data-blog.tistory.com/entry/Singular-Values-Decomposition-1>

Dekomposisi matriks menggunakan metode SVD dilakukan dengan beberapa langkah-langkah seperti menentukan nilai eigen dan vektor eigen dari matriks MM^T atau matriks M^TM . Vektor eigen dari matriks MM^T digunakan untuk membuat matriks U dan matriks M^TM untuk membuat matriks V . Nilai eigen dari salah satu matriks MM^T atau matriks M^TM yang telah dilakukan proses akar pangkat dua atau disebut nilai singular akan dibuat untuk membuat matriks Σ dengan susunan menurun diagonal.

B. Statistik Performa Pemain

Menurut KBBI, statistik adalah catatan angka-angka yang dikumpulkan, ditabulasi, digolong-golongkan sehingga dapat memberi informasi yang berarti mengenai suatu masalah atau gejala. Dalam konteks permainan basket, terdapat statistik yang didapatkan dari hasil pertandingan para pemain. Statistik tersebut sangat banyak jenisnya dan digunakan sesuai dengan kepentingannya masing-masing. Pada penelitian kali ini, penulis menggunakan statistik dari *traditional splits* yaitu *points*, *rebounds*, *assists*, *steals*, *blocks*, *turnovers*, persentase tembakan berhasil masuk, persentase tembakan tiga poin yang berhasil masuk, persentase tembakan bebas

yang berhasil masuk, dan +/-.

Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	
Steven Adams	C	26	OKC	58	58	27.0	4.5	7.6	.591	0.0590	3.4	6.0	9.4	2.4	0.9	1.1	1.5	1.9	10.9
Bam Adebayo	PF	22	MIA	65	65	34.4	6.3	11.1	.567	0.0690	2.5	8.0	10.5	5.1	1.2	1.3	2.8	2.5	16.2
LaMarcus Aldridge	C	34	SAS	53	53	33.1	7.4	15.0	.493	1.2827	1.9	5.5	7.4	2.4	0.7	1.6	1.4	2.4	18.9
Nickel Alexander-Walker	SG	21	NOP	41	0	12.2	1.9	5.5	.339	1.0607	0.2	1.8	2.0	1.8	0.3	0.2	1.0	1.1	5.1
Grayson Allen	SG	24	MEM	30	0	16.6	2.6	5.9	.449	1.1857	0.2	2.0	2.2	1.4	0.2	0.0	0.8	1.2	7.4
Thaddeus Young	PF	31	CHI	64	16	24.9	4.2	9.4	.448	1.2553	1.5	3.5	4.9	1.8	1.4	0.4	1.6	2.1	10.3
Trae Young	PG	21	ATL	60	60	35.3	9.1	20.8	.437	3.4880	0.5	3.7	4.3	9.3	1.1	0.1	4.8	1.7	29.6
Cody Zeller	C	27	CHO	58	39	23.1	4.3	8.3	.524	0.3682	2.8	4.3	7.1	1.5	0.7	0.4	1.3	2.4	11.1
Ante Zabic	C	23	CLE	22	0	10.0	1.9	3.3	.569	0.0737	0.8	2.2	3.0	0.3	0.3	0.2	0.5	1.2	4.4
Ivica Zubac	C	22	LAC	64	62	18.1	3.2	5.3	.601	0.0759	2.6	4.6	7.2	1.1	0.2	0.9	0.8	2.3	8.0

Gambar 2. Contoh Statistik Pemain
Sumber: <https://mussalains.com/?m=139339915>

Points adalah seberapa banyak pemain dapat memasukkan bola basket ke dalam ring lawan melalui tembakan dua angka, tiga angka, ataupun tembakan bebas. *Rebounds* adalah jumlah bola yang dapat diamankan seorang pemain dari tembakan yang gagal, baik itu saat menyerang ataupun saat bertahan. *Assists* adalah jumlah operan pemain kepada pemain lain yang langsung dapat menghasilkan poin. *Steals* adalah jumlah bola yang bisa direbut dari lawan saat posisi bertahan. *Blocks* adalah jumlah tembakan yang berhasil dihalangi oleh pemain saat posisi bertahan. persentase tembakan yang dibagi menjadi tiga yaitu tembakan dua poin, tiga poin, dan tembakan bebas dihitung dari jumlah tembakan yang masuk dibagi jumlah tembakan total dikali seratus persen. Hasil (+/-) adalah data yang mengukur kontribusi pemain saat berada di lapangan dimana (+) didapatkan bila saat pemain di lapangan dan timnya dapat menciptakan poin ke ring lawan dan (-) bila saat pemain di lapangan dan tim lawan dapat menciptakan poin ke ring miliknya.

C. Panjang (Norm) Vektor

Vektor adalah sebuah ruas garis yang memiliki arah dan besaran dengan arah tertentu. Panjang vektor atau *norm* didefinisikan sebagai jarak antara dua titik vektor. Panjang vektor memiliki beberapa rumus sebagai berikut.

Vektor	Vektor Posisi	Panjang Vektor
\vec{p}	$\vec{p} = (x, y)$	$ \vec{p} = \sqrt{x^2 + y^2}$
\vec{a} \vec{b}	$\vec{a} = (x_1, y_1)$ $\vec{b} = (x_2, y_2)$	$ \vec{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
\vec{q}	$\vec{q} = (x, y, z)$	$ \vec{q} = \sqrt{x^2 + y^2 + z^2}$
\vec{c} \vec{d}	$\vec{c} = (x_1, y_1, z_1)$ $\vec{d} = (x_2, y_2, z_2)$	$ \vec{CD} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$

Gambar 3. Rumus Panjang Vektor

Sumber: <https://idschool.net/sma/vektor-posisi-penyajian-vektor-dan-panjang-vektor/>

Rumus ini digunakan dalam penelitian ini untuk menghitung persentase kontribusi tiap pemain dalam sebuah tim.

IV. ANALISIS

Proses analisis yang dilakukan dalam penelitian kali ini melalui beberapa tahapan. Tahapan tersebut ditujukan untuk mendapatkan persentase kontribusi pemain dalam tim. Setidaknya ada empat tahap yaitu penyusunan data, penerapan SVD, identifikasi komponen utama, dan hitung kontribusi relatif.

A. Penyusunan Data

Data yang telah dikumpulkan pada penelitian kali ini sejumlah 16 pemain dengan sepuluh statistik tiap pemainnya ditambah dengan menit per pertandingan. Berikut visualisasi data yang sudah ada dalam bentuk tabel.

Player	Min	PTS	REB	AST	STL	BLK	TOV	FG%	3P%	FT%	+/-
Jordan Poole	30.1	21.0	2.3	4.9	1.9	0.5	3.4	44.8	42.1	88.4	-11.8
Kyle Kuzma	27.5	15.8	5.6	1.9	0.4	0.3	2.4	42.0	27.6	68.8	-12.5
Malcolm Brogdon	24.0	14.6	4.6	2.9	0.6	0.3	1.6	55.1	35.7	87.5	-9.3
Bilal Coulibaly	35.1	12.6	5.4	2.8	1.3	0.6	1.9	48.2	30.9	75.5	-14.1
Jonas Valanciunas	20.0	12.0	7.6	2.2	0.4	0.7	1.7	58.1	25.0	93.3	-5.4
Corey Kispert	28.7	11.3	3.5	1.4	0.7	0.1	1.2	41.7	31.5	81.8	-8.2
Alexandre Sarr	26.8	10.6	6.1	2.1	0.7	1.8	1.3	37.6	24.4	66.7	-12.9
Carlton Carrington	27.7	8.8	3.9	4.4	1.2	0.3	1.9	40.1	35.2	79.2	-12.2
Kyshawn George	27.0	7.9	4.0	2.7	0.7	0.7	1.1	34.0	25.3	73.1	-6.1
Jared Butler	7.8	4.3	0.6	2.2	0.5	0.2	0.5	46.5	16.7	71.4	1.0
Marin Bagley III	3.6	3.5	2.3	0.3	0.2	0.1	0.6	50.0	0.0	72.7	1.1
Richaun Holmes	3.9	3.0	1.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	0.0
Johnny Davis	7.1	2.6	1.2	0.2	0.3	0.1	0.1	48.3	50.0	60.0	1.5
Patrick Baldwin Jr.	5.1	2.3	1.1	0.1	0.2	0.2	0.2	47.4	45.5	50.0	2.5
Anthony Gill	5.1	2.1	0.9	0.4	0.2	0.1	0.1	58.8	50.0	53.8	1.0
Justin Champagnie	2.3	0.0	1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	6.0

Tabel 1. Tabel Visualisasi Data

Data tersebut kemudian disaring terkhusus untuk pemain yang memiliki rata-rata menit per pertandingan lebih besar dari rata-rata. Rata-rata menit per pertandingan dari data tersebut adalah 18.1 menit. Berikut visualisasi data yang sudah disaring.

Player	Min	PTS	REB	AST	STL	BLK	TOV	FG%	3P%	FT%	+/-
Jordan Poole	30.1	21.0	2.3	4.9	1.9	0.5	3.4	44.8	42.1	88.4	-11.8
Kyle Kuzma	27.5	15.8	5.6	1.9	0.4	0.3	2.4	42.0	27.6	68.8	-12.5
Malcolm Brogdon	24.0	14.6	4.6	2.9	0.6	0.3	1.6	55.1	35.7	87.5	-9.3
Bilal Coulibaly	35.1	12.6	5.4	2.8	1.3	0.6	1.9	48.2	30.9	75.5	-14.1
Jonas Valanciunas	20.0	12.0	7.6	2.2	0.4	0.7	1.7	58.1	25.0	93.3	-5.4
Corey Kispert	28.7	11.3	3.5	1.4	0.7	0.1	1.2	41.7	31.5	81.8	-8.2
Alexandre Sarr	26.8	10.6	6.1	2.1	0.7	1.8	1.3	37.6	24.4	66.7	-12.9
Carlton Carrington	27.7	8.8	3.9	4.4	1.2	0.3	1.9	40.1	35.2	79.2	-12.2
Kyshawn George	27.0	7.9	4.0	2.7	0.7	0.7	1.1	34.0	25.3	73.1	-6.1

Tabel 2. Tabel Visualisasi Data Setelah Disaring

Data tersebut akan dinormalisasi terlebih dahulu karena terdapat skala yang berbeda seperti pada *points* yang bisa bernilai puluhan sedangkan *steals* atau *block* hanya bernilai satuan. Normalisasi data tersebut distandarasi agar setiap kolom memiliki rata-rata nol dan standar deviasi satu menggunakan rumus *z-score* sebagai berikut.

$$z = \frac{x - \mu}{\sigma}$$

- z : Nilai *z-score* hasil normalisasi
- x : Nilai asli yang ingin dinormalisasi
- μ : Rata-rata dari dataset.
- σ : Standar deviasi dari dataset.

Data yang sudah dinormalisasi dimasukkan kembali ke dalam tabel dengan visualisasi sebagai berikut.

Nama	PTS	REB	AST	STL	BLK	TOV	FG%	3P%	FT%	+/-
Jordan Poole	2.20	-1.66	1.92	2.19	-0.19	-2.33	0.02	1.99	1.04	0.51
Kyle Kuzma	0.82	0.55	-0.84	-1.02	-0.61	-0.84	-0.35	-0.58	-1.22	0.75
Malcolm Brogdon	0.50	-0.12	0.08	-0.60	-0.61	0.35	1.40	0.86	0.94	-0.33
Bilal Coulibaly	-0.04	0.42	-0.01	0.91	0.02	-0.10	0.48	0.01	-0.45	1.29
Jonas Valanciunas	-0.20	1.90	-0.56	-1.02	0.24	0.20	1.80	-1.04	1.61	-1.65
Corey Kispert	-0.38	-0.86	-1.30	-0.38	-1.04	0.94	-0.39	0.11	0.28	-0.70
Alexandre Sarr	-0.57	0.89	-0.65	-0.38	2.58	0.79	-0.94	-1.14	-1.46	0.89
Carlton Carrington	-1.05	-0.59	1.46	0.69	-0.61	-0.10	-0.61	0.77	-0.02	0.65
Kyshawn George	-1.29	-0.52	-0.10	-0.38	0.24	1.09	-1.42	-0.98	-0.72	-1.41

Tabel 3. Visualisasi Data Setelah Normalisasi dengan Metode *z-score*

Data tersebut akan diubah formatnya menjadi matriks ukuran 16x10. Komponen kolom pada matriks meliputi data statistik yang telah dinormalisasi dan komponen baris pada matriks meliputi data tiap pemain. Visualisasi matriks terkait data yang sudah divisualisasi adalah sebagai berikut.

$$\begin{bmatrix}
 2.2 & -1.66 & 1.92 & 2.19 & -0.19 & -2.33 & 0.02 & 1.99 & 1.04 & 0.51 \\
 0.82 & 0.55 & -0.84 & -1.02 & -0.61 & -0.84 & -0.35 & -0.58 & -1.22 & 0.75 \\
 0.5 & -0.12 & 0.08 & -0.6 & -0.61 & 0.35 & 1.4 & 0.86 & 0.94 & -0.33 \\
 -0.04 & 0.42 & -0.01 & 0.91 & 0.02 & -0.1 & 0.48 & 0.01 & -0.45 & 1.29 \\
 -0.2 & 1.9 & -0.56 & -1.02 & 0.24 & 0.2 & 1.8 & -1.04 & 1.61 & -1.65 \\
 -0.38 & -0.86 & -1.3 & -0.38 & -1.04 & 0.94 & -0.39 & 0.11 & 0.28 & -0.7 \\
 -0.57 & 0.89 & -0.65 & -0.38 & 2.58 & 0.79 & -0.94 & -1.14 & -1.46 & 0.89 \\
 -1.05 & -0.59 & 1.46 & 0.69 & -0.61 & -0.1 & -0.61 & 0.77 & -0.02 & 0.65 \\
 -1.29 & -0.52 & -0.1 & -0.38 & 0.24 & 1.09 & -1.42 & -0.98 & -0.72 & -1.41
 \end{bmatrix}$$

Gambar 4. Matriks Ukuran 9x10 Berisi Data Setelah Normalisasi

B. Penerapan SVD

Matriks yang sudah berisi data pemain yang telah dinormalisasi kemudian dilakukan proses SVD dengan rumus yang telah disebutkan di atas. Hasil matriks Σ berukuran 9x9 untuk mempermudah perhitungan dengan matriks U . Hal tersebut tidak mempengaruhi perhitungan karena baris terakhir dari matriks Σ merupakan nilai 0. Berikut visualisasi dari hasil dekomposisi matriks tersebut dengan SVD.

$$U = \begin{bmatrix}
 0.79 & -0.05 & -0.11 & -0.15 & -0.46 & 0.04 & 0.09 & 0.04 & 0.33 \\
 -0.09 & -0.12 & -0.30 & 0.74 & -0.14 & -0.43 & -0.15 & 0.02 & 0.33 \\
 0.09 & 0.40 & 0.03 & 0.08 & 0.21 & 0.37 & -0.64 & -0.35 & 0.33 \\
 0.07 & -0.14 & -0.27 & 0.02 & 0.50 & 0.13 & 0.57 & -0.44 & 0.33 \\
 -0.27 & 0.68 & -0.26 & -0.33 & -0.10 & -0.28 & 0.21 & 0.20 & 0.33 \\
 -0.11 & 0.12 & 0.48 & 0.36 & -0.06 & 0.47 & 0.33 & 0.41 & 0.33 \\
 -0.38 & -0.49 & -0.40 & -0.30 & -0.15 & 0.36 & -0.21 & 0.24 & 0.33 \\
 0.19 & -0.21 & 0.30 & -0.22 & 0.55 & -0.40 & -0.21 & 0.39 & 0.33 \\
 -0.29 & -0.19 & 0.53 & -0.20 & -0.35 & -0.25 & 0.02 & -0.51 & 0.33
 \end{bmatrix}$$

Gambar 5. Matriks U

$$\Sigma = \begin{bmatrix}
 6.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
 0.00 & 4.57 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
 0.00 & 0.00 & 3.54 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
 0.00 & 0.00 & 0.00 & 2.70 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
 0.00 & 0.00 & 0.00 & 0.00 & 2.08 & 0.00 & 0.00 & 0.00 & 0.00 \\
 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.61 & 0.00 & 0.00 & 0.00 \\
 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.09 & 0.00 & 0.00 \\
 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.55 & 0.00 \\
 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00
 \end{bmatrix}$$

Gambar 6. Matriks Σ

$$V^T = \begin{bmatrix}
 0.44 & 0.27 & 0.32 & 0.35 & 0.29 & -0.43 & 0.30 & -0.09 & 0.28 \\
 0.13 & -0.46 & 0.38 & 0.15 & -0.38 & -0.19 & -0.36 & 0.50 & 0.18 \\
 -0.16 & 0.06 & -0.03 & 0.25 & 0.22 & 0.11 & -0.09 & 0.28 & -0.57 \\
 0.26 & 0.11 & -0.05 & 0.63 & -0.23 & 0.02 & -0.18 & -0.18 & -0.45 \\
 -0.17 & -0.22 & 0.50 & 0.06 & -0.31 &td>
 0.28 & 0.66 & -0.14 & -0.19 \\
 0.14 & 0.10 & 0.07 & -0.33 & -0.49 & -0.30 & -0.24 & -0.54 & -0.20 \\
 -0.25 & 0.05 & 0.32 & -0.29 & 0.25 & -0.58 & 0.01 & 0.15 & -0.43 \\
 0.35 & 0.05 & -0.49 & -0.18 & -0.33 & -0.22 & 0.47 & 0.44 & -0.18 \\
 -0.63 & 0.48 & -0.08 & 0.29 & -0.38 & -0.23 & 0.05 & 0.13 & 0.26 \\
 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00
 \end{bmatrix}$$

Gambar 7. Matriks V^T

C. Reduksi dengan Identifikasi Komponen Utama

Proses reduksi matriks disini bertujuan untuk membuat data lebih sederhana. Reduksi disini menggunakan metode variansi matriks yang mana hanya diambil 90-95% variansi data dari matriks. Persentase variansi ini dihitung dengan membagi tiap nilai singular kuadrat dibagi variansi total. Variansi total didapatkan dari jumlah semua singular kuadrat. Visualisasi rumusnya seperti ini.

$$\text{Persentase Variansi} = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^r \sigma_i^2}$$

k : Jumlah komponen yang harus dipilih

r : Jumlah nilai singular

Pada penelitian kali ini, jumlah komponen yang dipilih adalah empat karena berdasarkan nilai singular yang didapatkan, empat komponen telah mencapai nilai 90.65%. Dikarenakan komponen yang digunakan hanya empat, maka matriks U dan matriks Σ direduksi menjadi hanya memiliki empat kolom. Berikut visualisasi dari matriks U dan matriks Σ yang telah direduksi.

$$U = \begin{bmatrix} 0.79 & -0.05 & -0.11 & -0.15 \\ -0.09 & -0.12 & -0.30 & 0.74 \\ 0.09 & 0.40 & 0.03 & 0.08 \\ 0.07 & -0.14 & -0.27 & 0.02 \\ -0.27 & 0.68 & -0.26 & -0.33 \\ -0.11 & 0.12 & 0.48 & 0.36 \\ -0.38 & -0.49 & -0.40 & -0.30 \\ 0.19 & -0.21 & 0.30 & -0.22 \\ -0.29 & -0.19 & 0.53 & -0.20 \end{bmatrix}$$

Gambar 8. Matriks U Setelah Reduksi

$$\Sigma = \begin{bmatrix} 6.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 4.57 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 3.54 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 2.70 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

Gambar 9. Matriks Σ Setelah Reduksi

Setelah proses reduksi, dilakukan operasi perkalian matriks antara matriks U dan matriks Σ yang telah direduksi. Perkalian kedua matriks tersebut akan menghasilkan matriks A proyeksi. Matriks ini akan diproses ke tahap selanjutnya. Visualisasi dari hasil perkalian kedua matriks ini adalah sebagai berikut.

$$A = \begin{bmatrix} 5.0481 & -0.2285 & -0.3894 & -0.405 & 0 & 0 & 0 & 0 \\ -0.5751 & -0.5484 & -1.062 & 1.998 & 0 & 0 & 0 & 0 \\ 0.5751 & 1.828 & 0.1062 & 0.216 & 0 & 0 & 0 & 0 \\ 0.4473 & -0.6398 & -0.9558 & 0.054 & 0 & 0 & 0 & 0 \\ -1.7253 & 3.1076 & -0.9204 & -0.891 & 0 & 0 & 0 & 0 \\ -0.7029 & 0.5484 & 1.6992 & 0.972 & 0 & 0 & 0 & 0 \\ -2.4282 & -2.2393 & -1.416 & -0.81 & 0 & 0 & 0 & 0 \\ 1.2141 & -0.9597 & 1.062 & -0.594 & 0 & 0 & 0 & 0 \\ -1.8531 & -0.8683 & 1.8762 & -0.54 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Gambar 10. Matriks A Proyeksi Setelah Reduksi

D. Hitung Kontribusi Relatif

Hasil dari matriks A proyeksi dihitung dengan kontribusi relatif. Kontribusi relatif dihitung dengan panjang vektor per baris dari matriks A proyeksi dibagi dengan total panjang vektor dari semua baris vektor. Visualisasi rumus dari kontribusi relatif ini adalah sebagai

berikut.

$$\text{Kontribusi Relatif} = \frac{|p_i|}{\sum_{j=1}^m |p_j|} \times 100\%$$

$|p_i|$: Panjang vektor pemain per pemain

$\sum_{j=1}^m |p_j|$: Total panjang vektor dari semua pemain

Setelah proses ini dilakukan maka akan didapatkan persentase kontribusi pemain terhadap tim dan divisualisasikan dengan tabel sebagai berikut.

Player	Contribution (%)
Jordan Poole	14.14
Kyle Kuzma	14.08
Malcolm Brogdon	7.27
Bilal Coulibaly	5.43
Jonas Valančiūnas	14.71
Corey Kispert	10.82
Alexandre Sarr	13.84
Carlton Carrington	8.20
Kyshawn George	11.52

Tabel 4. Visualisasi Hasil Kontribusi Tiap Pemain

Berdasarkan hasil visualisasi tersebut, persentase kontribusi pemain yang dihasilkan oleh metode *singular value decomposition* (SVD) sesuai dengan statistik tiap pemain dari Washington Wizards. Pemain dengan kontribusi tertinggi sebesar 14.71% yaitu Jonas Valančiūnas. Berdasarkan statistik yang ada, Jonas Valančiūnas memiliki poin per pertandingan cukup tinggi, lalu memiliki persentase tembakan masuk yang cukup tinggi juga, selain itu komponen bertahan seperti *rebounds*, *steals*, dan *blocks* yang bisa dibilang cukup tinggi daripada pemain lainnya. Pemain dengan persentase kontribusi paling rendah adalah Bilal Coulibaly dengan persentase kontribusi sebesar 5.43%. Berdasarkan statistik yang ada, Bilal Coulibaly memang memiliki beberapa komponen statistik yang cukup tinggi, namun ia sangat kurang pada komponen (+/-) yaitu -14. Hal itu mengindikasikan bahwa saat dia di lapangan maka Washington Wizards rata-rata kehilangan 14 poin.

Analisis lainnya dari hasil persentase performa tiap pemain Washington Wizards di atas, bisa dilihat bahwa terdapat variasi persentase kontribusi sembilan pemain Washington Wizards dengan menit bermain di atas rata-rata. Hal tersebut mungkin menjadi salah satu kurangnya performa tim Washington Wizards yang sangat kurang dibandingkan tim-tim lainnya.

V. IMPLEMENTASI

Pada penelitian kali ini, penulis melakukan implementasi penelitian ini dengan program berbahasa python. Program dibuat sesuai dengan tahapan analisis yang telah dijelaskan diatas dengan sedikit penambahan fitur untuk mempermudah proses pemasukan data. Berikut kode implementasi tersebut yang akan dijelaskan per bagian.

A. Input Data

```
import numpy as np
import pandas as pd

# 1. Memuat Data dari File CSV
PathFile = r"D:\Algeo\Makalah\Cavs_Stats.csv"
DataFrame = pd.read_csv(PathFile)

# Ambil nama pemain dan statistik
NamaPemain = DataFrame["Player"].values
Statistik = DataFrame.columns[1:]

# Filter data: hanya pemain dengan menit per pertandingan di atas rata-rata
RataRataMenit = DataFrame["Min"].mean()
DataFrameTersaring = DataFrame[DataFrame["Min"] > RataRataMenit]

# Kolom yang digunakan untuk matriks
KolomDipilih = ["PTS", "REB", "AST", "STL", "BLK", "TOV", "FG%", "3P%", "FT%", "+/-"]
NamaPemain = DataFrameTersaring["Player"].values
Statistik = KolomDipilih
DataTersaring = DataFrameTersaring[KolomDipilih].values

# Membalik tanda untuk statistik buruk (TOV dan +/- negatif jadi positif)
DataTersaring[:, KolomDipilih.index("TOV")] *= -1
```

Gambar 11. Bagian Kode Input Data

Pada kode tersebut, kode menerima input *file* dalam format csv. File yang sudah dimasukkan akan diambil nama pemain dan statistiknya. Setelah itu, data akan disaring berdasarkan pemain yang memiliki rata-rata menit bermain diatas rata-rata menit bermain timnya. Kemudian data tersebut akan disaring lagi terkhusus untuk komponen statistik *points*, *rebounds*, *assists*, *steals*, *blocks*, *turnovers*, persentase tembakan berhasil masuk, persentase tembakan tiga poin yang berhasil masuk, persentase tembakan bebas yang berhasil masuk, dan hasil (+/-). Tidak lupa juga untuk mengalikan komponen TOV dengan -1 karena TOV merupakan salah satu komponen negatif.

B. Normalisasi Data

```
# Normalisasi Data (z-score normalization)
RataRataData = np.mean(DataTersaring, axis=0)
StandarDeviasiData = np.std(DataTersaring, axis=0)
DataTernormalisasi = (DataTersaring - RataRataData) / StandarDeviasiData

# Menampilkan data hasil matriks normalisasi
DataFrameNormalisasi = pd.DataFrame(
    DataTernormalisasi, index=NamaPemain, columns=Statistik)
print("Data setelah normalisasi (z-score):")
print(DataFrameNormalisasi.to_string(float_format="%.2f"))
```

Gambar 12. Bagian Kode Normalisasi Data

Pada kode tersebut, data yang sudah melewati tahap penyaringan akan dinormalisasi. Normalisasi data tersebut menggunakan metode *z-score*. Data yang sudah dinormalisasi akan ditampilkan dalam terminal dalam bentuk matriks. Berikut *output* yang dihasilkan.

Data setelah normalisasi (z-score):

	PTS	REB	AST	STL	BLK	TOV	FG%	3P%	FT%	+/-
Donovan Mitchell	1.73	0.02	0.67	1.81	-0.54	-1.04	-0.45	0.33	0.51	-0.02
Darius Garland	1.21	-0.74	2.25	0.87	-0.54	-1.68	0.04	0.73	1.31	0.07
Evan Mobley	0.83	1.54	-0.07	0.56	2.25	-1.17	0.73	-0.17	0.57	1.29
Jarrett Allen	0.26	2.01	-0.64	-0.07	1.32	0.24	2.30	-2.62	-0.02	-0.63
Caris LeVert	-0.20	-0.61	0.84	-0.07	-0.08	0.11	0.40	0.95	0.12	1.95
Georges Niang	-0.73	-0.34	-0.92	-1.64	-0.78	0.24	-0.70	-0.14	-0.24	-0.49
Sam Merrill	-0.93	-0.97	-0.75	-1.32	-0.78	1.01	-1.05	0.13	0.86	-1.24
Isaac Okoro	-1.05	-0.77	-0.64	-0.38	-0.54	1.14	-0.17	0.74	-0.85	-1.15
Dean Wade	-1.12	-0.14	-0.75	0.24	-0.31	1.14	-1.09	0.05	-2.27	0.21

Gambar 13. Output Terminal Normalisasi Data

C. Dekomposisi SVD

```
# Dekomposisi SVD
MatriksU, NilaiSingular, MatriksVT = np.linalg.svd(
    DataTernormalisasi, full_matrices=False)

# Menampilkan matriks hasil dekomposisi SVD
print("\nMatriks U:")
print(pd.DataFrame(MatriksU, index=NamaPemain,
    columns=[f"PC{i+1}" for i in range(MatriksU.shape[1])]))

print("\nMatriks Sigma (Nilai Singular):")
print(pd.DataFrame(np.diag(NilaiSingular), columns=[
    f"PC{i+1}" for i in range(len(NilaiSingular))],
    index=[f"PC{i+1}" for i in range(len(NilaiSingular))]))

print("\nMatriks VT:")
print(pd.DataFrame(MatriksVT, columns=Statistik,
    index=[f"PC{i+1}" for i in range(MatriksVT.shape[0])]))
```

Gambar 14. Bagian Kode Dekomposisi SVD

Pada kode tersebut, data yang sudah dinormalisasi akan didekomposisi menggunakan matriks SVD. Matriks hasil dekomposisi akan ditampilkan ke terminal. Hasil dari matriks pada visualisasi akan terdapat beberapa perbedaan, seperti matriks Σ yang berukuran 9x9 dan matriks V^T yang berukuran 9x10. Hal itu dikarenakan *library* python yang digunakan menggunakan memori lebih efisien pada programnya. Walaupun demikian, hal ini tidak mempengaruhi perhitungan. Hasil *output* dari kode tersebut adalah sebagai berikut.

Matriks U:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Donovan Mitchell	0.30	0.30	-0.08	0.57	-0.23	0.07	-0.47	0.30	-0.33
Darius Garland	0.36	0.50	-0.28	0.01	0.28	0.07	0.51	-0.30	-0.33
Evan Mobley	0.46	-0.25	0.27	-0.28	-0.58	-0.29	0.20	-0.03	-0.33
Jarrett Allen	0.24	-0.74	-0.29	0.13	0.40	0.11	-0.10	-0.06	-0.33
Caris LeVert	0.08	0.21	0.43	-0.53	0.43	0.09	-0.41	0.13	-0.33
Georges Niang	-0.32	-0.04	-0.18	-0.22	-0.25	0.60	0.28	0.46	-0.33
Sam Merrill	-0.39	0.08	-0.45	-0.24	-0.25	-0.18	-0.38	-0.47	-0.33
Isaac Okoro	-0.37	0.02	-0.00	0.14	0.22	-0.67	0.25	0.42	-0.33
Dean Wade	-0.35	-0.08	0.58	0.42	-0.02	0.21	0.13	-0.44	-0.33

Gambar 15. Output Terminal Matriks U

Matriks Sigma:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
PC1	6.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PC2	0.00	5.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PC3	0.00	0.00	3.23	0.00	0.00	0.00	0.00	0.00	0.00
PC4	0.00	0.00	0.00	2.58	0.00	0.00	0.00	0.00	0.00
PC5	0.00	0.00	0.00	0.00	1.83	0.00	0.00	0.00	0.00
PC6	0.00	0.00	0.00	0.00	0.00	1.26	0.00	0.00	0.00
PC7	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00
PC8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00
PC9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Gambar 16. Output Terminal Matriks Σ

Matriks VT:

	PTS	REB	AST	STL	BLK	TOV	FG%	3P%	FT%	+/-
PC1	0.44	0.27	0.32	0.35	0.29	-0.43	0.30	-0.09	0.28	0.26
PC2	0.13	-0.46	0.38	0.15	-0.38	-0.19	-0.36	0.50	0.18	0.09
PC3	-0.16	0.06	-0.03	0.25	0.22	0.11	-0.09	0.28	-0.57	0.66
PC4	0.26	0.11	-0.05	0.63	-0.23	0.02	-0.18	-0.18	-0.45	-0.45
PC5	-0.17	-0.22	0.50	0.06	-0.31	0.28	0.66	-0.14	-0.19	0.02
PC6	0.14	0.10	0.07	-0.33	-0.49	-0.30	-0.24	-0.54	-0.20	0.38
PC7	-0.25	0.05	0.32	-0.29	0.25	-0.58	0.01	0.15	-0.43	-0.37
PC8	0.35	0.05	-0.49	-0.18	-0.33	-0.22	0.47	0.44	-0.18	0.00
PC9	-0.63	0.48	-0.08	0.29	-0.38	-0.23	0.05	0.13	0.26	0.04

Gambar 17. Output Terminal Matriks V^T

D. Identifikasi Komponen Utama Relevan

```
# Identifikasi komponen utama yang relevan
VariansiDijelaskan = (NilaiSingular ** 2) / np.sum(
    NilaiSingular ** 2)
VariansiKumulatif = np.cumsum(VariansiDijelaskan)

print(
    "\nVariansi yang dijelaskan oleh setiap komponen ut
    ma:"
)
for i, (Var, VarKum) in enumerate(zip(
    VariansiDijelaskan, VariansiKumulatif), start=1):
    print(f"Komponen {i}: {Var:.2%} (kumulatif: {
    VarKum:.2%})")

# Pilih jumlah komponen utama yang relevan
JumlahKomponenRelevan = np.argmax(VariansiKumulatif
    >= 0.9) + 1
print(f"\nJumlah komponen utama yang relevan: {
    JumlahKomponenRelevan}")
```

Gambar 18. Bagian Kode Perhitungan Komponen Utama

Kode tersebut akan melakukan identifikasi komponen utama yang relevan. Komponen utama yang dipilih adalah komponen dengan persentase diatas 90%. Jika sudah sampai di 90%, maka proses pencarian akan berhenti dan menggunakan jumlah komponen utama yang telah dihitung. Kode ini juga akan mengeluarkan *output* ke terminal. Berikut hasil *output* ke terminal.

Variansi yang dijelaskan oleh setiap komponen utama:

- Komponen 1: 43.23% (kumulatif: 43.23%)
- Komponen 2: 30.78% (kumulatif: 74.01%)
- Komponen 3: 11.57% (kumulatif: 85.58%)
- Komponen 4: 7.37% (kumulatif: 92.95%)
- Komponen 5: 3.73% (kumulatif: 96.68%)
- Komponen 6: 1.76% (kumulatif: 98.44%)
- Komponen 7: 1.01% (kumulatif: 99.45%)
- Komponen 8: 0.55% (kumulatif: 100.00%)
- Komponen 9: 0.00% (kumulatif: 100.00%)

Jumlah komponen utama yang relevan: 4

Gambar 19. Output Terminal Komponen Utama

E. Perhitungan Kontribusi

```
# Visualisasi kontribusi relatif pemain
MatriksURINGkas = MatriksU[:, :JumlahKomponenRelevan
]
Kontribusi = np.linalg.norm(MatriksURINGkas, axis=1)
KontribusiRelatif = (Kontribusi / np.sum(Kontribusi
)) * 100

# Membuat DataFrame untuk tabel kontribusi
TabelKontribusi = pd.DataFrame({
    "Nama Pemain": NamaPemain,
    "Kontribusi (%)": KontribusiRelatif
})

# Menampilkan tabel kontribusi relatif
print(
    "\nTabel Kontribusi Relatif Pemain (Total 100%):")
print(TabelKontribusi.to_string(index=False,
    float_format="% .2f"))
```

Gambar 20. Bagian Kode Perhitungan Kontribusi Pemain

Kode tersebut akan menghitung persentase kontribusi tiap pemain yang. Kode tersebut juga akan mengeluarkan output kontribusi pemain dalam terminal. Berikut hasil *output* kode ke terminal.

Tabel Kontribusi Relatif Pemain (Total 100%):	
Nama Pemain	Kontribusi (%)
Donovan Mitchell	12.23
Darius Garland	11.48
Evan Mobley	11.10
Jarrett Allen	14.28
Caris LeVert	12.19
Georges Niang	7.32
Sam Merrill	11.06
Isaac Okoro	6.75
Dean Wade	13.59

Gambar 20. Output Kontribusi Pemain

VI. KESIMPULAN

Berdasarkan penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa *Singular Value Decomposition* (SVD) bisa digunakan untuk analisis olahraga. Pada penelitian kali ini, penulis

menggunakannya untuk menganalisis performa tiap pemain dari tim Washington Wizards dengan rata-rata menit bermain diatas rata-rata dan pembatasan data pada musim 2024/2025 sebelum bulan Desember 2024. Hal ini tidak menutup kemungkinan bahwa data lain ataupun tim lain bisa digunakan. Selain itu, tahapan analisis yang sama dengan sedikit modifikasi bisa digunakan untuk menganalisis data dari olahraga lainnya.

Penulis berharap penelitian lanjutan yang lebih mendetail baik itu terkait analisis ataupun data statistik bisa dikembangkan suatu hari. Dengan demikian, SVD bisa menjadi sebuah metode yang berguna bagi pelatih, analisis olahraga, dan juga sebagai salah satu metode pengambilan keputusan.

VII. LAMPIRAN

Link Repository Kode:

https://github.com/rakadaffa/player_performance

Link Video:

<https://www.youtube.com/watch?v=AERphndOos>

VIII. UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena dengan nikmat dan anugerahnya penulis bisa menyelesaikan makalah ini. Penulis juga menyampaikan terima kasih kepada Dr. Ir. Rinaldi Munir T. sebagai dosen pengampu IF2123 Aljabar Linier dan Geometri kelas K02 atas bimbingan dan pemberian ilmu sebagai dasar penulisan makalah ini. Penulis juga berterima kasih kepada ibu penulis karena tanpa dukungan dari beliau, penulis belum tentu bisa menyelesaikan makalah ini. Terakhir, penulis juga mengucapkan terima kasih kepada teman-teman penulis serta pihak-pihak lain yang telah mendorong penulis dalam menyelesaikan makalah ini.

REFERENSI

- Adha, Sophia Maulidatul. "Vektor Matematika – Pengertian, Rumus, dan Contoh Soal." *Aku Pintar*, 2022, <https://akupintar.id/info-pintar/-/blogs/vektor-matematika-pengertian-rumus-dan-contoh-soal>. Diakses pada 26 Desember 2024 pukul 14.40 WIB.
- Ariyanti, Gregoria. "Dekomposisi Nilai Singular dan Aplikasinya." *Seminar Nasional Matematika dan Pendidikan Matematika*, 2010, pp. 33-39. [core.ac.uk](https://core.ac.uk/download/pdf/11067339.pdf), <https://core.ac.uk/download/pdf/11067339.pdf>.
- Elha, Maulia Mumtaz. "Mengenal Sejarah NBA, Kompetisi Basket Paling Bergengsi." *JakartaNotebook*, 8 September 2023, <https://www.jakartanotebook.com/blog/sejarah-nba-hbl/>. Diakses pada 25 Desember 2024 pukul 10.01 WIB.

"NBA Advanced Stats." *The official site of the NBA for the latest NBA Scores, Stats & News.* | *NBA.com*, 2024, <http://nba.com>. Diakses 27 Desember 2024 pukul 12.39 WIB.

PERNYATAAN

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

Bandung, 29 Desember 2024



Raka Daffa Iftikhaar - 13523018