

Penerapan Dekomposisi QR untuk Optimasi Penempatan Panel Surya pada Smart Building

Boye Mangaratua Ginting - 13523127¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹boyeginting@gmail.com, 13523127@std.stei.itb.ac.id

Abstract— Dekomposisi QR merupakan metode matematis yang dapat diimplementasikan untuk mengoptimalkan penempatan panel surya pada smart building dengan mempertimbangkan berbagai faktor kritis seperti efek bayangan, orientasi panel, dan batasan struktur bangunan. Tingkat efektivitas metode ini terlihat dari kemampuannya menganalisis interaksi kompleks antar panel surya dan menghasilkan solusi optimal yang memaksimalkan produksi energi. Melalui penerapan dekomposisi QR, posisi dan orientasi panel surya dapat diatur sedemikian rupa sehingga meminimalkan interferensi bayangan antar panel sambil memaksimalkan paparan sinar matahari yang diterima. Di balik optimasi penempatan panel surya ini, terdapat prinsip-prinsip aljabar linear yang diterapkan untuk menganalisis matriks interaksi bayangan dan menghasilkan konfigurasi optimal yang meningkatkan efisiensi sistem secara keseluruhan..

Keywords— dekomposisi QR, panel surya, smart building, optimasi, energi terbarukan.

I. PENDAHULUAN

Di era modern ini, konsep *smart building* telah menjadi solusi penting dalam menghadapi tantangan efisiensi energi dan keberlanjutan lingkungan. Smart building merupakan integrasi sistem otomatisasi, teknologi informasi, dan manajemen energi yang bertujuan untuk mengoptimalkan penggunaan sumber daya sambil meningkatkan kenyamanan penghuni. Salah satu komponen kunci dalam konsep ini adalah pemanfaatan energi terbarukan, khususnya melalui implementasi sistem panel surya.

Panel surya, sebagai teknologi konversi energi matahari menjadi listrik, memiliki peran vital dalam mewujudkan bangunan yang mandiri energi. Namun, efektivitas sistem panel surya sangat bergantung pada optimasi penempatannya. Faktor-faktor seperti sudut kemiringan, arah hadap, dan efek bayangan antar panel dapat secara signifikan mempengaruhi efisiensi pembangkitan energi. Tantangan utama dalam optimasi penempatan panel surya adalah kompleksitas perhitungan yang melibatkan berbagai variabel dan konstrain.

Dalam konteks ini, metode dekomposisi QR menawarkan pendekatan matematis yang menjanjikan. Dekomposisi QR, sebuah teknik dalam aljabar linear yang memecah matriks menjadi komponen ortogonal (Q) dan triangular atas (R), dapat digunakan untuk menganalisis independensi posisi panel dan mengoptimalkan penempatan untuk meminimalkan interferensi bayangan. Metode ini memungkinkan analisis sistematis terhadap interaksi antar panel dan pengaruhnya terhadap efisiensi sistem secara keseluruhan.

Pembahasan pada makalah ini akan mengeksplorasi bagaimana dekomposisi QR dapat diterapkan dalam optimasi penempatan panel surya pada *smart building*, dengan mempertimbangkan berbagai aspek teknis dan praktis. Pembahasan akan mencakup formulasi matematis dari masalah, implementasi algoritma, dan analisis hasil dalam konteks aplikasi nyata. Tujuan akhirnya adalah memberikan *framework* yang dapat digunakan untuk meningkatkan efisiensi sistem panel surya pada *smart building* melalui optimasi penempatan yang tepat.

II. LANDASAN TEORI

2.1 Dekomposisi QR

Dekomposisi QR atau faktorisasi QR adalah metode dalam aljabar linear yang memecah sebuah matriks A menjadi hasil kali dari matriks Q (orthogonal) dan matriks R (upper triangular). Secara matematis, untuk sebuah matriks A berukuran $m \times n$, dekomposisi QR dapat dinyatakan sebagai:

$$A = QR$$

dimana Q adalah matriks orthogonal berukuran $m \times m$ ($Q^T Q = Q Q^T = I$) dan R adalah matriks segitiga atas (upper triangular) berukuran $m \times n$.

Contoh dekomposisi QR

$$\begin{matrix} A & & Q & & R \\ \begin{pmatrix} 2.5 & 1.1 & 0.3 \\ 2.2 & 1.9 & 0.4 \\ 1.8 & 0.1 & 0.3 \end{pmatrix} & = & \begin{pmatrix} -0.7 & 0.1 & -0.7 \\ -0.6 & -0.7 & 0.4 \\ -0.5 & 0.7 & 0.5 \end{pmatrix} & \begin{pmatrix} -3.8 & -1.9 & -0.6 \\ 0 & -1.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \end{matrix}$$

Gambar 2.1.1: Contoh dekomposisi QR

2.1.1 Sifat-Sifat Matriks Orthogonal

Matriks Q dalam dekomposisi QR memiliki beberapa sifat penting, yakni kolom-kolomnya orthonormal (saling tegak lurus dan memiliki panjang 1), $Q^T Q = Q Q^T = I$ (inverse matriksnya sama dengan transposenya), dan $\|Qx\| = \|x\|$ untuk semua vektor x (mempertahankan panjang).

2.1.2 Metode Gram-Schmidt

Proses Gram-Schmidt adalah salah satu metode utama untuk melakukan dekomposisi QR. Prosesnya mengubah himpunan vektor bebas linear menjadi himpunan vektor orthonormal.

Untuk vektor a_1, a_2, \dots, a_n , proses Gram-Schmidt menghasilkan vektor orthonormal q_1, q_2, \dots, q_n dengan:

1. $q_1 = a_1/|a_1|$
2. Untuk $k > 1$: $\tilde{u}_k = a_k - \sum_{j=1}^{k-1} \text{proj}_{q_j}(a_k)$ $q_k = \tilde{u}_k/|\tilde{u}_k|$

dimana $\text{proj}_{q_j}(a_k) = (a_k \cdot q_j)q_j$

2.1.3 Algoritma Householder

Metode Householder adalah alternatif yang lebih stabil secara numerik dibandingkan Gram-Schmidt. Metode ini menggunakan refleksi Householder untuk mengubah matriks A menjadi bentuk segitiga atas.

Refleksi Householder didefinisikan sebagai:

$$H = I - \frac{2vv^T}{v^Tv}$$

dimana, v adalah vector Householder yang dipilih untuk menghasilkan nol pada elemen-elemen yang diinginkan.

2.1.4 Aplikasi dalam Optimasi

Dalam konteks optimasi penempatan panel surya, dekomposisi QR digunakan untuk :

1. Analisis independensi, dengan matriks Q menunjukkan arah-arah independen untuk penempatan panel, dan nilai-nilai diagonal R mengindikasikan tingkat ketergantungan antar posisi.
2. Minimalisasi interferensi, dimana orthogonalitas Q membantu posisi yang meminimalkan bayangan dan struktur triangular R digunakan untuk mempermudah analisis hierarkis pengaruh antar panel.
3. Optimasi numerik, yakni dekomposisi QR lebih stabil dalam stabilitas numeriknya jika dibandingkan dengan metode langsung seperti eliminasi Gauss, dan memiliki efisiensi komputasi yang lebih baik dalam menyelesaikan sistem persamaan linear.

2.1.5 Hasil Optimasi

- Koordinat optimal setiap panel
- Sudut kemiringan optimal
- Minimalisasi efek bayangan
- Maksimalisasi paparan sinar matahari

2.1.6 Kompleksitas Komputasi

Kompleksitas waktu untuk dekomposisi QR:

- Metode Gram-Schmidt: $O(mn^2)$
- Metode Householder: $O(mn^2)$

dimana $m \times n$ adalah dimensi matriks input.

III. PEMBAHASAN

3.1 Formulasi Masalah Optimasi

Optimasi penempatan panel surya dapat diformulasikan sebagai masalah minimalisasi interferensi bayangan antar panel dengan mempertimbangkan berbagai konstrain. Secara matematis, masalah ini dapat direpresentasikan sebagai berikut:

$$\min f(P) = \sum_{ij} S(p_i, p_j)$$

dengan batasan:

- $x_{\min} \leq x_i \leq x_{\max}$
- $y_{\min} \leq y_i \leq y_{\max}$
- $|\theta_i| \leq 90^\circ$

dimana:

- $P = \{p_i\}$ adalah himpunan posisi panel dengan $p_i = (x_i, y_i, \theta_i)$
- $S(p_i, p_j)$ adalah fungsi efek bayangan antara panel i dan j
- $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ adalah batas area instalasi panel surya

3.2 Representasi Matriks Bayangan

Interaksi bayangan antar panel direpresentasikan dalam matriks bayangan S, dimana setiap elemen S_{ij} merepresentasikan efek bayangan dari panel i terhadap panel j . Matriks ini memiliki karakteristik:

- Dimensi $n \times n$, dimana n adalah jumlah panel
- $S_{ii} = 0$ (tidak ada efek bayangan panel terhadap dirinya sendiri)
- $0 \leq S_{ij} \leq 1$ (efek bayangan dinormalisasi)

3.3 Penerapan Dekomposisi QR

Dekomposisi QR digunakan untuk menganalisis dan mengoptimalkan penempatan panel melalui langkah-langkah berikut:

3.3.1 Dekomposisi Matriks Bayangan Matriks bayangan S didekomposisi menjadi: $S = QR$ dimana:

- Q adalah matriks orthogonal yang merepresentasikan arah-arah independen
- R adalah matriks segitiga atas yang merepresentasikan hierarki pengaruh bayangan

Contoh inputan posisi awal panel surya:

$$\begin{bmatrix} 0 & 0 & 80 \\ 2 & 0 & 30 \\ 0 & 2 & 30 \end{bmatrix}$$

Gambar 3.3.1.1: Contoh input posisi awal panel surya

dimana, kolom pertama menunjukkan posisi panel pada x , kolom kedua menunjukkan posisi panel pada y , dan kolom ketiga menunjukkan sudut panel.

3.3.2 Analisis Komponen Q

Kolom-kolom matriks Q memberikan informasi tentang:

• **Independensi Bayangan Antar Panel:**
Setiap kolom $Q[:, i]$ merupakan vektor basis yang menggambarkan efek bayangan suatu panel dalam ruang interaksi bayangan.

Jika kolom-kolom Q saling tegak lurus, ini menunjukkan bahwa interaksi bayangan antar panel dapat dianggap independen.

- **Dimensi Relatif Bayangan:**

Besar nilai setiap kolom Q memberikan gambaran tentang kontribusi bayangan suatu panel terhadap sistem secara keseluruhan.

- Transformasi Stabil:

Q melakukan transformasi yang stabil terhadap sistem, memungkinkan representasi efek bayangan tanpa memengaruhi hubungan antar elemen.

3.3.3 Interpretasi Matriks R

Elemen-elemen diagonal R memberikan informasi tentang:

- Interaksi Dominan Antar Panel:

Elemen diagonal $R[i, i]$ menunjukkan kontribusi bayangan langsung dari panel i . Nilai yang besar menunjukkan bahwa panel i memiliki efek bayangan yang signifikan terhadap sistem.

- Interaksi Antar Panel:

Elemen di luar diagonal, ($R[i, j]$, dengan $i \neq j$) merepresentasikan pengaruh bayangan panel i terhadap panel j .

Semakin kecil elemen-elemen ini, semakin lemah interaksi bayangan antar panel.

- Kekuatan Sistem Secara Keseluruhan:

Nilai diagonal yang besar secara kolektif mengindikasikan sistem yang kurang optimal, karena banyak panel saling menghalangi. Sebaliknya, nilai diagonal kecil menunjukkan konfigurasi yang efisien.

3.3.4 Alur Perhitungan Optimasi

Inputan merupakan tebakan awal yang dapat diatur secara acak dalam penempatan posisi panel surya. Inputan hanya digunakan untuk menghitung tingkat optimasi yang akan terjadi.

Kemudian matriks bayangan (A) dibuat berdasarkan posisi panel dan posisi matahari yang sudah ditentukan. Kemudian dekomposisi QR dilakukan pada matriks A :

$$A = Q \cdot R$$

dan fungsi objektif dihitung menggunakan elemen diagonal $R = [i, i]$:

$$\text{Fungsi Objektif} : f(x) = \sum_{i=1}^n |R[i, i]|$$

Kemudian digunakan algoritma optimasi *Sequential Least Squares Programming (SLSQP)* untuk meminimalkan fungsi objektif $f(x)$. Selama proses iterasi berlangsung, posisi dan sudut panel diperbarui untuk meminimalkan nilai $f(x)$, dengan memperhitungkan batas area (x, y) dan sudut panel. Proses iterasi akan berhenti ketika nilai fungsi objektif sudah tidak berubah secara signifikan atau telah mencapai nilai minimum.

3.4 Implementasi Algoritma

Implementasi algoritma optimasi menggunakan pendekatan berikut:

3.4.1 Inisialisasi

Python

```
def create_shadow_matrix(panel_positions,
sun_positions):
"""
Membentuk matriks bayangan berdasarkan
posisi panel dan posisi matahari
"""
```

Fungsi ini menginisialisasi matriks bayangan dengan mempertimbangkan:

- Posisi relatif antar panel
- Posisi matahari sepanjang hari
- Sudut kemiringan panels

3.4.2 Fungsi Objektif

Python

```
def objective(positions):
# Create shadow interaction matrix
shadow_matrix =
create_shadow_matrix(panel_configs,
sun_positions)
# Apply QR decomposition
Q, R = np.linalg.qr(shadow_matrix)
```

Fungsi objektif mengevaluasi kualitas penempatan dengan:

- Membentuk matriks interaksi bayangan
- Melakukan dekomposisi QR
- Menggunakan elemen diagonal R sebagai metrik kualitas

3.4.3 Optimasi

Python

```
result = minimize(
objective,
initial_positions.flatten(),
constraints=constraints,
method='SLSQP'
)
```

Proses optimasi menggunakan metode SLSQP (Sequential Least Squares Programming) dengan:

- Konstrain area instalasi
- Konstrain sudut panel
- Minimalisasi efek bayangan total

3.5 Analisis Hasil

Hasil optimasi memberikan:

- Posisi optimal (x, y) untuk setiap panel
- Sudut kemiringan optimal (θ) untuk setiap panel
- Minimalisasi interferensi bayangan antar panel
- Konfigurasi yang memenuhi semua konstrain area dan sudut

3.6 Evaluasi Performa

Performa algoritma dapat dievaluasi berdasarkan:

- Waktu komputasi: $O(n^3)$ untuk n panel
- Konvergensi: Bergantung pada kondisi awal dan

kompleksitas konstrain

- Stabilitas numerik: Dijamin oleh penggunaan dekomposisi QR
- Skalabilitas: Efektif untuk sistem hingga puluhan panel

IV. PROGRAM DAN EKSPERIMEN

```
import numpy as np
from scipy.optimize import minimize

def create_shadow_matrix(panel_positions,
sun_positions):
    n_panels = len(panel_positions)
    shadow_matrix = np.zeros((n_panels,
n_panels))

    for i in range(n_panels):
        for j in range(n_panels):
            if i != j:
                shadow_effect = 0
                for sun_pos in sun_positions:
                    shadow =
calculate_shadow_effect(
                        panel_positions[i],
                        panel_positions[j],
                        sun_pos
                    )
                    shadow_effect += shadow
                shadow_matrix[i,j] =
shadow_effect

    return shadow_matrix

def calculate_shadow_effect(panel1, panel2,
sun_position):
    x1, y1, angle1 = panel1
    x2, y2, angle2 = panel2
    azimuth, elevation = sun_position

    panel_normal = np.array([
        np.sin(np.radians(angle1)),
        np.cos(np.radians(angle1)),
        0
    ])
    sun_vector = np.array([
        np.cos(np.radians(elevation)) *
np.cos(np.radians(azimuth)),
        np.cos(np.radians(elevation)) *
np.sin(np.radians(azimuth)),
        np.sin(np.radians(elevation))
    ])

    effective_height = 2 *
abs(np.dot(panel_normal, sun_vector))
    shadow_length = effective_height /
np.tan(np.radians(elevation))
```

```
        shadow_x = x1 + shadow_length *
np.cos(np.radians(azimuth))
        shadow_y = y1 + shadow_length *
np.sin(np.radians(azimuth))

        dist = np.sqrt((shadow_x - x2)**2 +
(shadow_y - y2)**2)

        panel2_normal = np.array([
            np.sin(np.radians(angle2)),
            np.cos(np.radians(angle2)),
            0
        ])
        angle_factor = abs(np.dot(panel2_normal,
sun_vector))

        shadow_effect = max(0, 1 -
dist/shadow_length) * angle_factor

    return shadow_effect

def calculate_objective(positions,
sun_positions):
    panel_configs = positions.reshape(-1, 3)
    shadow_matrix =
create_shadow_matrix(panel_configs,
sun_positions)

    Q, R = np.linalg.qr(shadow_matrix)
    quality_metric =
np.sum(np.abs(np.diag(R)))

    avg_elevation = np.mean(sun_positions[:,
1])
    optimal_tilt = 90 - avg_elevation # Sudut
optimal relatif terhadap elevasi matahari

    angle_penalty = 0
    for panel in panel_configs:
        angle_diff = abs(panel[2] -
optimal_tilt)
        angle_penalty += angle_diff

    return quality_metric + 0.5 *
angle_penalty

def
optimize_panel_placement(initial_positions,
sun_positions, area_constraints):
    constraints = [
        {'type': 'ineq', 'fun': lambda x:
area_constraints['max_x'] - x[:,3]},
        {'type': 'ineq', 'fun': lambda x:
x[:,3] - area_constraints['min_x']},
        {'type': 'ineq', 'fun': lambda x:
area_constraints['max_y'] - x[1::3]},
```

```

        {'type': 'ineq', 'fun': lambda x:
x[1::3] - area_constraints['min_y']},
        {'type': 'ineq', 'fun': lambda x: 60
- np.abs(x[2::3])}, # Membatasi sudut maksimum
ke 60 derajat
        {'type': 'ineq', 'fun': lambda x:
x[2::3] + 10}, # Membatasi sudut minimum ke -
10 derajat
    ]

    result = minimize(
        lambda x: calculate_objective(x,
sun_positions),
        initial_positions.flatten(),
        constraints=constraints,
        method='SLSQP',
        options={'ftol': 1e-8, 'maxiter':
1000}
    )

    return result.x.reshape(-1, 3)

if __name__ == "__main__":
    initial_positions = np.array([
        [0, 0, 80],
        [2, 0, 20],
        [0, 2, 30],
        [2, 2, 10]
    ])

    sun_positions = np.array([
        [90, 30], # Morning
        [180, 60], # Noon
        [270, 30] # Afternoon
    ])

    area_constraints = {
        'min_x': 0,
        'max_x': 4,
        'min_y': 0,
        'max_y': 4
    }

    optimal_positions =
optimize_panel_placement(
        initial_positions,
        sun_positions,
        area_constraints
    )

    print("\nInitial panel positions (x, y,
angle):")
    print(initial_positions)
    print("\nOptimal panel positions (x, y,
angle):")
    print(optimal_positions)

```

```

    initial_objective =
calculate_objective(initial_positions.flatten(
), sun_positions)
    final_objective =
calculate_objective(optimal_positions.flatten(
), sun_positions)

    print("\nOptimization improvement:")
    print(f"Initial objective value:
{initial_objective:.4f}")
    print(f"Final objective value:
{final_objective:.4f}")
    print(f"Improvement:
{((initial_objective - final_objective) /
initial_objective * 100):.2f}%")

```

Dan untuk hasil optimasi, dapat dilihat contohnya pada gambar di bawah ini.

```

Initial panel positions (x, y, angle):
[[ 0  0 80]
 [ 2  0 20]
 [ 0  2 30]
 [ 2  2 10]]

Optimal panel positions (x, y, angle):
[[1.50550611e+00 1.30029654e-02 5.33472207e+01]
 [2.31916584e+00 2.27930990e-02 4.52193547e+01]
 [1.10430874e-02 3.56537537e+00 3.31582522e+01]
 [9.22200640e-01 3.99057096e+00 1.35551748e+01]]

Optimization improvement:
Initial objective value: 80.9150
Final objective value: 31.1458
Improvement: 61.51%

```

Gambar 4.1: Contoh Hasil Optimasi

V. KESIMPULAN

Metode dekomposisi QR dalam program ini digunakan untuk menilai dan mengurangi interaksi bayangan antar panel secara sistematis. Dengan meminimalkan elemen diagonal matriks R, panel surya ditempatkan sedemikian rupa sehingga meminimalkan bayangan antar mereka, yang pada akhirnya meningkatkan efisiensi energi. Pembahasan dan program juga masih dapat diperluas dengan menambahkan parameter seperti efisiensi energi aktual berdasarkan sudut atau intensitas matahari untuk optimasi yang lebih sesuai dengan yang ada di lapangan.

VII. UCAPAN TERIMA KASIH

Dengan penuh rasa syukur dan kerendahan hati, penulis ingin menyampaikan ucapan terima kasih kepada Tuhan Yang Maha Esa dan semua pihak yang berkontribusi dalam penyelesaian makalah ini. Terima kasih kepada Bapak Dr. Judhi Santoso, M.Sc dan Bapak Arrival Dwi Sentosa, S.Kom., M.T, selaku dosen pengampu mata kuliah IF2123 Aljabar Linier dan

Geometri K03 yang sudah membimbing penulis dan teman-teman selama perkuliahan.

Terima kasih juga kepada pembaca yang sudah meluangkan waktunya untuk membaca makalah ini. Semoga makalah ini dapat memberikan ide dan sudut pandang yang baru dalam bidang penerapan ilmu aljabar linier dan geometri.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-23b-Dekomposisi-QR-2024.pdf>, diakses pada 29 Desember 2024
- [2] Duffie, J. A., & Beckman, W. A. (2013). Solar engineering of thermal processes (4th ed.). Wiley.
- [3] Golub, G. H., & Van Loan, C. F. (2013). Matrix computations (4th ed.). Johns Hopkins University Press.
- [4] Nocedal, J., & Wright, S. J. (2006). Numerical optimization (2nd ed.). Springer.

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, 1 Januari 2025



Boye Mangaratua Ginting
13523127