

## DAFTAR PUSTAKA

- Armitage, P., Berry, G., & Matthews, J.N.S. (2002). *Statistical Methods in Medical Research*. 4th ed. Blackwell Science. <https://doi.org/10.1002/9780470773666>
- Franke, T., Ho, T., & Christie, Christina A., The Chi-Square Test. (2012). *American Journal of Evaluation*. <https://doi.org/10.1177/1098214011426594>
- Gulic, M., Maglic, L., Krljan, T., dan Maglic, L. (2022). Solving the Container Relocation Problem by Using a Metaheuristic Genetic Algorithm. *Applied Sciences*. 12(7397), 1-5 (2022). <https://doi.org/10.3390/app12157397>
- Kurniawati, F. (2015). Studi Kelayakan Pengembangan Terminal Peti Kemas Pelabuhan Belang-belang. *Warta Penelitian Perhubungan* Vol. 27 No. 3. (2015). <https://doi.org/10.25104/warlit.v27i3.780>
- Maglic, L., Gulic, M., dan Maglic, L. (2020). Optimization Of Container Relocation Operations In Port Container Terminal. *Transport*. 35(1), 37-47 (2020). <https://doi.org/10.3846/transport.2019.11628>
- Montgomery, D.C. (2012). *Design and Analysis of Experiments*. 8th ed. Hoboken, NJ: John Wiley & Sons. <https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/181-Design-and-Analysis-of-Experiments-Douglas-C.-Montgomery-Edisi-8-2013.pdf>
- Permenhub RI. (2016). *Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 116 Tahun 2016 Pasal 2(1)*. Permenhub. Jakarta
- Putra, A. A., Djalante, S. (2016). Pengembangan Infrastruktur Pelabuhan Dalam Mendukung Pembangunan Berkelanjutan. *Jurnal Ilmiah Media Engineering Vol.6*. (2016). <https://ejournal.unsrat.ac.id/index.php/jime/article/view/11627>
- Rana, Rakash & Singhal R., Chi-square test and its application in hypothesis testing. (2015). *Journal of the Practice of Cardiovascular Science*. <https://doi.org/10.4103/2395-5414.157577>.
- Sofwan, A., Handoyo, E., WD. R., (2008) Algoritma Genetika Dalam Pemilihan Spesifikasi Komputer. *Seminar Nasional Aplikasi Teknologi Informasi* (2008). <https://journal.uui.ac.id/Snati/article/view/176>
- Suryani, E., Hendrawan, R. A., Rahmawati, U. E., Cahyandini, G. A., Andika, M., Riski, R. (2021). Pendekatan Systems Thinking Peningkatan Pangsa Pasar Dan Profitabilitas Guna Meningkatkan Pendapatan UMKM Frozen. *Jurnal Direktorat Riset dan Pengabdian Kepada Masyarakat* (2021). <https://journal.its.ac.id/index.php/sewagati/article/view/400/305>

- Ting, C. dan Wu, K. (2017). Optimizing container relocation operations at container yards with beam search. *Transportation Research Part E*. 103 (1), 17-31 (2017). <https://doi.org/10.1016/j.tre.2017.04.010>
- Zhang, C., Guan, H., Yuan, Y., Chen, W., dan Wu, T. (2020). Machine learning-driven algorithms for the container relocation problem. *Transportation Research Part B*. 139(1), 102-131 (2020). <https://doi.org/10.1016/j.trb.2020.05.017>
- Zurkanain, R. (2022). *Apa itu Dwelling Time?* URL: <https://www.pajakonline.com/apa-itu-dwelling-time/>.

## **LAMPIRAN**

### Lampiran 1. Dataset

- [Terminal In & Out\\_2023 JUNI.xlsx](#)
- [Terminal In & Out\\_2023 april \(1\).xlsx](#)
- [Terminal In & Out\\_2023 maret \(1\).xlsx](#)
- [Terminal In & Out\\_2023 JULI.xlsx](#)
- [Terminal In & Out\\_2023 februari \(1\).xlsx](#)
- [Terminal In & Out\\_2023 october \(1\).xlsx](#)
- [Terminal In & Out mei 2023 \(1\).xlsx](#)
- [Terminal In & Out\\_2023 desember \(1\).xlsx](#)
- [Terminal In & Out\\_2023 januari \(1\).xlsx](#)
- [Terminal In & Out\\_2023 november \(1\).xlsx](#)

- [Container Truck Turn Time\\_IMPORT MAR \(1\).xlsx](#)
- [Container Truck Turn Time\\_JUNI complete...xlsx](#)
- [Container Truck Turn Time\\_IMPORT JUNI.xlsx](#)
- [Container Truck Turn Time\\_IMPORT APR \(1\).xlsx](#)
- [Container Truck Turn Time\\_MARET complete.xlsx](#)
- [Container Truck Turn Time\\_MEI complete.xlsx](#)
- [Container Truck Turn Time\\_JANUARI complete.xlsx](#)
- [Container Truck Turn Time\\_IMPORT MEI \(2\).xlsx](#)
- [Container Truck Turn Time\\_IMPORT JANUARI \(1\).xlsx](#)
- [Container Truck Turn Time\\_IMPORT FEB \(1\).xlsx](#)
- [Container Truck Turn Time\\_FEBRUARI complete.xlsx](#)
- [Container Truck Turn Time\\_APRIL complete.xlsx](#)

Terminal In & Out\_2023 JUNI

Container Truck Turn Time\_JUNI complete...

(a) Data mentah dari Politeknik Ilmu Pelayaran

Com	Leng	In Carrier ID	Out Carrier ID	In Terminal	Truck In	Truck Out	DW Time	DW Time (Days)	TrT Truck	Waiting Handling [TrT - 6]
GE	40	MRML-0002	0512	2023-05-29 01:35:00	2023-06-01 00:59:00	2023-06-01 01:26:00	2 days 23:51:00	5,145833333	0 days 00:27:00	27
GE	40	ID04-0041	1111	2023-05-31 16:24:00	2023-06-01 02:25:00	2023-06-01 03:43:00	0 days 11:19:00	5,611805556	0 days 01:18:00	78

(b) Pre-processing data : Menggabung semua data mentah dan menghapus data yang tidak diperlukan

Bay	No.	Container Priority	Time Spent (minutes)	Comm.	In Terminal	Truck In	Truck Out	DW Time	DW Time (Days)	DW Time (minutes)	TrT Truck	TrT Truck (Minutes)	Waiting Handling
1	1	1	1	6	GE	2023-01-26 00:13:00	2023-02-01 08:51:00	2023-02-01 08:08:00	6 days 07:55:00	6,329861111	9115	0 days 01:00:00	77
2	1	2	2	6	GE	2023-01-26 00:20:00	2023-02-02 10:15:00	2023-02-02 11:02:00	7 days 10:42:00	7,445833333	10722	0 days 00:00:00	47

(c) Data yang digunakan pada algoritma dan analisis perbandingan

## Lampiran 2. Algoritma Genetika

```

import matplotlib.pyplot as plt
import numpy as np
import random

def generate_matrix_with_user_input(rows, cols, fill_count):
    data = np.zeros((rows, cols))

    print("Silakan masukkan angka untuk container dalam matriks 4x6 dari kiri ke kanan setiap baris secara berurutan:")

    for row in range(rows - 1, -1, -1): # Start from the bottom row to the top row
        for col in range(cols):
            if fill_count == 0:
                break
            while True:
                try:
                    value = int(input(f"Masukkan container untuk kotak di baris {row + 1}, kolom {col + 1}: "))
                    data[row, col] = value
                    fill_count -= 1
                    break
                except ValueError:
                    print("Input tidak valid. Masukkan angka yang valid.")
            if fill_count == 0:
                break

    # Create plot

    fig, ax = plt.subplots()

    # Plot each box with its random color if it has a number
    for i in range(rows):
        for j in range(cols):
            if data[i, j] != 0: # Only color boxes with numbers
                color = np.random.rand(3,) # Generate a random color
                ax.add_patch(plt.Rectangle((j, i), 1, 1, color=color))
                ax.text(j + 0.5, i + 0.5, int(data[i, j]), va='center', ha='center', color='white')
            else:
                ax.add_patch(plt.Rectangle((j, i), 1, 1, color='white', edgecolor='white')) # Color empty boxes white

    # Adjust plot to look like a matrix
    ax.set_xlim(0, cols)
    ax.set_ylim(0, rows)
    ax.set_xticks(np.arange(cols) + 0.5)
    ax.set_yticks(np.arange(rows) + 0.5)
    ax.set_xticklabels(np.arange(cols))
    ax.set_yticklabels(np.arange(rows))

    ax.tick_params(top=False, bottom=False, left=False, right=False, labelleft=True, labelbottom=True)
    ax.grid(False)

    plt.gca().invert_yaxis()
    plt.show()

    # Display the position and value of each filled box
    for row in range(rows):
        for col in range(cols):
            if data[row, col] != 0:
                print(f"bay[{row}][{col}]={int(data[row, col])}")

# Set number of rows and columns
rows, cols = 4, 6
fill_count = int(input("Masukkan jumlah container yang diisi: "))
generate_matrix_with_user_input(rows, cols, fill_count)

```

(a) Input posisi kontainer

```

import numpy as np
import random
import time

# Definisikan parameter GA
POPULATION_SIZE = 100
CHROMOSOME_LENGTH = 66
ELITISM_FACTOR = 0.5
CROSSOVER_RATE = 0.6
MUTATION_RATE = 0.001
NUMBER_OF_EVOLUTIONS = 50

# Definisikan heuristik sebagai gen
HEURISTICS = [
    1: "TLP", # Ke Posisi Terendah (TLP)
    2: "RI", # Indeks Perpindahan
    3: "RIL", # Indeks Perpindahan Terendah
    4: "RI_TLP", # Indeks Perpindahan + TLP
    5: "RI_THP", # Indeks Perpindahan + THP
    6: "TLP_RI" # Ke Posisi Terendah + RI
]

# Inisialisasi bay dengan prioritas kontainer
def bay_init():
    bay = np.zeros((4, 6), dtype=int)
    bay[1][0] = 13
    bay[1][1] = 14
    bay[1][2] = 15
    bay[1][3] = 16
    bay[1][4] = 17
    bay[1][5] = 18
    bay[2][0] = 7
    bay[2][1] = 8
    bay[2][2] = 9
    bay[2][3] = 10
    bay[2][4] = 11
    bay[2][5] = 12
    bay[3][0] = 1
    bay[3][1] = 2
    bay[3][2] = 3
    bay[3][3] = 4
    bay[3][4] = 5
    bay[3][5] = 6
    return bay

```

(b) Inisialisasi parameter algoritma dan inisialisasi bay kontainer

```

# Visualisasikan kondisi bay saat ini
def visualize_bay(bay, iteration):
    print(f"\n{iteration} - Kondisi bay:")
    for row in bay:
        print(" ".join(f"{cell:2}" for cell in row))
    print("\n" + "-"*25)

# Fungsi fitness: menghitung jumlah pengocokan ulang yang diperlukan
def fitness_function(chromosome, detailed_moves=False, visualize=False):
    bay = bay_init().copy()
    reshuffles = 0
    moves_log = [] # Daftar untuk menyimpan detail pergerakan
    iteration = 0

    # Simulasikan proses pengambilan berdasarkan urutan heuristik dari kromosom
    for priority in range(1, 19): # prioritas dari 1 hingga 18
        position = np.argwhere(bay == priority)
        if position.size > 0:
            row, col = position[0]
            while row > 0: # Pindahkan kontainer yang memblokir kontainer prioritas
                top_container_position = np.argwhere(bay[:, col] > 0)
                if top_container_position.size > 0:
                    top_row = top_container_position[0][0]
                    if top_row < row: # Ada kontainer yang memblokir
                        # Pilih heuristik berdasarkan kromosom
                        heuristic = chromosome[iteration % len(chromosome)]

# Terapkan heuristik yang dipilih
if heuristic == 1: # TLP
    new_col, new_row = find_lowest_position(bay, col)
elif heuristic == 2: # RI
    new_col, new_row = find_reshuffle_index(bay, col, top_row)
elif heuristic == 3: # RIL
    new_col, new_row = find_reshuffle_index_lowest(bay, col, top_row)
elif heuristic == 4: # RI_TLP
    new_col, new_row = find_reshuffle_index_then_lowest(bay, col, top_row)
elif heuristic == 5: # RI_THP
    new_col, new_row = find_reshuffle_high_priority(bay, col, top_row)
elif heuristic == 6: # TLP_RI
    new_col, new_row = find_lowest_position_then_reshuffle(bay, col)

moves_log.append(f"Pindahkan kontainer {bay[top_row][col]} dari kolom {col} ke kolom {new_col}")
bay[new_row][new_col] = bay[top_row][col]
bay[top_row][col] = 0
reshuffles += 1
iteration += 1

        if visualize:
            visualize_bay(bay, iteration)
            time.sleep(1)
        else:
            break # Tidak ada kontainer yang memblokir

# Hapus kontainer dari bay (diatur menjadi 0)
moves_log.append(f"Kontainer {priority} diambil dari kolom {col}")
bay[row][col] = 0

# Pindahkan kontainer lainnya ke bawah jika perlu
for r in range(row, 0, -1):
    bay[r][col] = bay[r-1][col]
    bay[r-1][col] = 0

iteration += 1
if visualize:
    visualize_bay(bay, iteration)
    time.sleep(1)

if detailed_moves:
    for move in moves_log:
        print(move)

return reshuffles, moves_log

```



```

# Fungsi pembantu untuk menemukan posisi terendah untuk memindahkan kontainer
def find_lowest_position(bay, current_col):
    for col in range(bay.shape[1]):
        if col != current_col:
            # Temukan posisi kosong terendah di kolom
            empty_rows = np.where(bay[:, col] == 0)[0]
            if len(empty_rows) > 0:
                return col, empty_rows[-1] # Kembalikan posisi kosong terendah
    return current_col, 0

def find_resuffle_index(bay, current_col, top_row):
    # For demonstration purposes, randomly select a column
    new_col = random.choice([c for c in range(bay.shape[1]) if c != current_col])
    empty_rows = np.where(bay[:, new_col] == 0)[0]
    if len(empty_rows) > 0:
        return new_col, empty_rows[-1]
    return current_col, 0

```

```

def find_resuffle_index_lowest(bay, current_col, top_row):
    # Pick column with the least containers (highest empty rows)
    min_height = bay.shape[0]
    target_col = current_col
    for col in range(bay.shape[1]):
        if col != current_col:
            height = np.count_nonzero(bay[:, col])
            if height < min_height:
                min_height = height
                target_col = col
    empty_rows = np.where(bay[:, target_col] == 0)[0]
    if len(empty_rows) > 0:
        return target_col, empty_rows[-1]
    return current_col, 0

def find_resuffle_index_then_lowest(bay, current_col, top_row):
    # Attempt reshuffle by RI then move to lowest
    col, row = find_resuffle_index(bay, current_col, top_row)
    return find_lowest_position(bay, col)

def find_resuffle_high_priority(bay, current_col, top_row):
    # Move to column with highest priority container (for demo, just random)
    new_col = random.choice([c for c in range(bay.shape[1]) if c != current_col])
    empty_rows = np.where(bay[:, new_col] == 0)[0]
    if len(empty_rows) > 0:
        return new_col, empty_rows[-1]
    return current_col, 0

def find_lowest_position_then_resuffle(bay, current_col):
    # First try lowest position then apply reshuffle index logic
    col, row = find_lowest_position(bay, current_col)
    return find_resuffle_index(bay, col, row)

# Proses seleksi berdasarkan fitness
def select_population(population):
    sorted_population = sorted(population, key=lambda x: fitness_function(x)[0])
    return sorted_population[:int(ELITISM_FACTOR * POPULATION_SIZE)]

```

```

# Operasi crossover
def crossover(parent1, parent2):
    if random.random() < CROSSOVER_RATE:
        crossover_point = random.randint(1, CHROMOSOME_LENGTH - 1)
        child1 = parent1[:crossover_point] + parent2[crossover_point:]
        child2 = parent2[:crossover_point] + parent1[crossover_point:]
        return child1, child2
    return parent1, parent2

# Operasi mutasi
def mutate(chromosome):
    for i in range(CHROMOSOME_LENGTH):
        if random.random() < MUTATION_RATE:
            chromosome[i] = random.randint(1, len(HEURISTICS))
    return chromosome

# Eksekusi Algoritma Genetika
def genetic_algorithm():
    population = generate_initial_population()
    for _ in range(NUMBER_OF_EVOLUTIONS):
        selected_population = select_population(population)
        next_generation = selected_population.copy()

        # Crossover dan Mutasi
        while len(next_generation) < POPULATION_SIZE:
            parent1, parent2 = random.sample(selected_population, 2)
            child1, child2 = crossover(parent1, parent2)
            next_generation.append(mutate(child1))
            if len(next_generation) < POPULATION_SIZE:
                next_generation.append(mutate(child2))

        population = next_generation

    # Kembalikan solusi terbaik
    best_solution = min(population, key=lambda x: fitness_function(x)[0])
    best_fitness, best_moves_log = fitness_function(best_solution, detailed_moves=True, visualize=False)
    return best_solution, best_fitness, best_moves_log

# Hasilkan populasi awal secara acak
def generate_initial_population():
    return [[random.randint(1, len(HEURISTICS)) for _ in range(CHROMOSOME_LENGTH)] for _ in range(POPULATION_SIZE)]

```

```

# Latih algoritma 10 kali dan simpan hasilnya
def train_multiple_times(num_trials=10):
    results = []
    for trial in range(num_trials):
        print(f"\nRunning trial {trial + 1}...")
        best_solution, best_fitness, best_moves_log = genetic_algorithm()
        results.append((best_solution, best_fitness, best_moves_log))
        print(f"Trial {trial + 1} - Best fitness: {best_fitness}")
    return results

# Jalankan Algoritma Genetika 20 kali
results = train_multiple_times(20)

# Temukan trial dengan fitness terendah
best_trial = min(results, key=lambda x: x[1])
best_solution, best_fitness, best_moves_log = best_trial

# Tampilkan perpindahan detail untuk trial dengan fitness terendah
print("\nPerpindahan detail untuk trial dengan fitness terendah:")
for move in best_moves_log:
    print(move)

# Tampilkan solusi dan fitness terbaik
print("\nSummary of results:")
for i, (solution, fitness, _) in enumerate(results, start=1):
    print(f"Trial {i}: Fitness {fitness} - Solution {solution}")

```

(c) Program Algoritma Genetika

```

def update_bay(bay, originalX, newX):
    for y in range(len(bay)):
        if bay[y][originalX] != 0:
            container = bay[y][originalX]
            bay[y][originalX] = 0
            for y in range(len(bay) - 1, -1, -1):
                if bay[y][newX] == 0:
                    bay[y][newX] = container
                    return 1.5 # Time spent moving the container
            return 0.0

def is_blocked(bay, y, x):
    return any(bay[i][x] != 0 for i in range(y))

def remove_top_priority(bay, priority, time_spent, removal_times, bay_iteration, get_container_number):
    for y in range(bay.shape[0] - 1, -1, -1):
        for x in range(bay.shape[1]):
            if bay[y][x] == priority and not is_blocked(bay, y, x):
                bay[y][x] = 0
                time_spent += 1.5 # Time spent removing the container
                removal_times.append((get_container_number(priority, bay_iteration), priority, time_spent - sum(t for _, _, t in removal_times)))
    return time_spent

def get_container_number1(priority, bay_iteration):
    if priority <= 6:
        return 115 + (bay_iteration - 20) * 6 + (priority - 1)
    elif 7 <= priority <= 12:
        return 307 + (bay_iteration - 20) * 6 + (priority - 7)
    elif 13 <= priority <= 18:
        return 499 + (bay_iteration - 20) * 6 + (priority - 13)
    return None

def get_container_number2(priority, bay_iteration):
    if priority <= 6:
        return priority + (bay_iteration - 1) * 6
    elif 7 <= priority <= 12:
        return 193 + (bay_iteration - 1) * 6 + (priority - 7)
    elif 13 <= priority <= 18:
        return 385 + (bay_iteration - 1) * 6 + (priority - 13)
    elif 19 <= priority <= 20:
        return 577 + (bay_iteration - 1) * 6 + (priority - 19)
    return None

# Define the list of moves based on input

```

✓ 2s completed at 2:40 PM

```

# Define the list of moves based on input
moves1 = [(0, 1), (0, 2), (1, 0), (1, 0), (2, 0), (2, 1), (2, 1), (3, 2), (3, 2), (4, 3), (4, 3), (5, 4), (5, 4), (0, 5)]
moves2 = [(0, 2), (0, 3), (0, 4), (1, 0), (1, 0), (1, 0), (2, 1), (2, 1), (2, 1), (3, 2), (3, 2), (3, 2), (4, 1), (4, 3), (4, 5), (5, 3), (5, 4), (5, 4), (2, 5)]

# Initialize DataFrames to accumulate results
all_removal_times1 = pd.DataFrame(columns=["Bay", "No.", "Container Priority", "Time Spent (minutes)"])
all_removal_times2 = pd.DataFrame(columns=["Bay", "No.", "Container Priority", "Time Spent (minutes)"])

# Simulation 1
for bay_iteration in range(20, 33):
    bay = bay_init1()
    removal_times = []
    total_time = 0.0
    priorities = list(range(1, 19))
    move_copy = moves1.copy()

    for priority in priorities:
        while True:
            total_time = remove_top_priority(bay, priority, total_time, removal_times, bay_iteration, get_container_number1)
            if any(removal[1] == priority for removal in removal_times):
                break
            if move_copy:
                originalX, newX = move_copy.pop(0)
                total_time += update_bay(bay, originalX, newX)

    removal_df = pd.DataFrame(removal_times, columns=["No.", "Container Priority", "Time Spent (minutes)"])
    removal_df["Bay"] = bay_iteration
    all_removal_times1 = pd.concat([all_removal_times1, removal_df], ignore_index=True)

# Simulation 2
for bay_iteration in range(1, 20):
    bay = bay_init2()
    removal_times = []
    total_time = 0.0
    priorities = list(range(1, 21))
    move_copy = moves2.copy()

```

```

for priority in priorities:
    while True:
        total_time = remove_top_priority(bay, priority, total_time, removal_times, bay_iteration, get_container_number2)
        if any(removal[1] == priority for removal in removal_times):
            break
        if move_copy:
            originalX, newX = move_copy.pop(0)
            total_time += update_bay(bay, originalX, newX)

        removal_df = pd.DataFrame(removal_times, columns=["No.", "Container Priority", "Time Spent (minutes)"])
        removal_df["Bay"] = bay_iteration
        all_removal_times2 = pd.concat([all_removal_times2, removal_df], ignore_index=True)

# Combine both results
combined_removal_times = pd.concat([all_removal_times1, all_removal_times2], ignore_index=True)

# Export the combined DataFrame to an Excel file
output_path = 'combined_container_removal_times.xlsx'
combined_removal_times.to_excel(output_path, index=False)

# Use Google Colab's file system to download the Excel file
from google.colab import files
files.download(output_path)

```

(d) Keluarkan waktu pengambilan setiap kontainer

## Lampiran 3. Hasil Algoritma

```

Running trial 1...
Pindahkan kontainer 19 dari kolom 0 ke kolom 2
Pindahkan kontainer 13 dari kolom 0 ke kolom 3
Pindahkan kontainer 7 dari kolom 0 ke kolom 4
Kontainer 1 diambil dari kolom 0
Pindahkan kontainer 20 dari kolom 1 ke kolom 5
Pindahkan kontainer 14 dari kolom 1 ke kolom 0
Pindahkan kontainer 8 dari kolom 1 ke kolom 0
Kontainer 2 diambil dari kolom 1
Pindahkan kontainer 19 dari kolom 2 ke kolom 1
Pindahkan kontainer 15 dari kolom 2 ke kolom 1
Pindahkan kontainer 9 dari kolom 2 ke kolom 0
Kontainer 3 diambil dari kolom 2
Pindahkan kontainer 13 dari kolom 3 ke kolom 0
Pindahkan kontainer 16 dari kolom 3 ke kolom 2
Pindahkan kontainer 10 dari kolom 3 ke kolom 3
Pindahkan kontainer 10 dari kolom 3 ke kolom 2
Kontainer 4 diambil dari kolom 3
Pindahkan kontainer 7 dari kolom 4 ke kolom 3
Pindahkan kontainer 17 dari kolom 4 ke kolom 4
Pindahkan kontainer 17 dari kolom 4 ke kolom 1
Pindahkan kontainer 11 dari kolom 4 ke kolom 4
Pindahkan kontainer 11 dari kolom 4 ke kolom 3
Kontainer 5 diambil dari kolom 4
Pindahkan kontainer 20 dari kolom 5 ke kolom 4
Pindahkan kontainer 18 dari kolom 5 ke kolom 1
Pindahkan kontainer 12 dari kolom 5 ke kolom 2
Kontainer 6 diambil dari kolom 5
Pindahkan kontainer 11 dari kolom 3 ke kolom 5
Kontainer 7 diambil dari kolom 3
Pindahkan kontainer 9 dari kolom 0 ke kolom 3
Pindahkan kontainer 13 dari kolom 0 ke kolom 3
Kontainer 8 diambil dari kolom 0
Pindahkan kontainer 13 dari kolom 3 ke kolom 0
Kontainer 9 diambil dari kolom 3
Pindahkan kontainer 12 dari kolom 2 ke kolom 4
Kontainer 10 diambil dari kolom 2
Kontainer 11 diambil dari kolom 5
Kontainer 12 diambil dari kolom 4
Kontainer 13 diambil dari kolom 0
Kontainer 14 diambil dari kolom 0
Pindahkan kontainer 18 dari kolom 1 ke kolom 0
Pindahkan kontainer 17 dari kolom 1 ke kolom 2
Kontainer 15 diambil dari kolom 1
Pindahkan kontainer 17 dari kolom 2 ke kolom 0
Kontainer 16 diambil dari kolom 2
Kontainer 17 diambil dari kolom 0
Kontainer 18 diambil dari kolom 0
Trial 1 - Best fitness: 29

```

## (a) Hasil iterasi 1 pada algoritma

```

Summary of results:
Trial 1: Fitness 29 - Solution [1, 1, 3, 4, 4, 3,
Trial 2: Fitness 23 - Solution [5, 1, 1, 2, 4, 1,
Trial 3: Fitness 21 - Solution [2, 3, 3, 3, 5, 1,
Trial 4: Fitness 24 - Solution [4, 3, 2, 2, 1, 1,
Trial 5: Fitness 25 - Solution [3, 3, 1, 3, 5, 3,
Trial 6: Fitness 21 - Solution [3, 3, 1, 6, 3, 6,
Trial 7: Fitness 27 - Solution [1, 1, 6, 3, 3, 1,
Trial 8: Fitness 22 - Solution [1, 3, 6, 2, 3, 3,
Trial 9: Fitness 24 - Solution [4, 1, 3, 2, 3, 1,
Trial 10: Fitness 21 - Solution [1, 3, 3, 4, 3, 1,
Trial 11: Fitness 25 - Solution [3, 1, 2, 5, 1, 6,
Trial 12: Fitness 23 - Solution [1, 3, 1, 5, 3, 4,
Trial 13: Fitness 19 - Solution [1, 3, 3, 5, 4, 1,
Trial 14: Fitness 27 - Solution [3, 2, 3, 1, 3, 4,
Trial 15: Fitness 25 - Solution [2, 1, 3, 6, 1, 1,
Trial 16: Fitness 21 - Solution [4, 3, 1, 6, 1, 3,
Trial 17: Fitness 27 - Solution [1, 3, 6, 6, 1, 3,
Trial 18: Fitness 21 - Solution [4, 3, 3, 6, 2, 3,
Trial 19: Fitness 22 - Solution [1, 1, 1, 2, 1, 4,
Trial 20: Fitness 23 - Solution [4, 3, 1, 2, 2, 3,

```

## (b) Hasil fitness seluruh 20 iterasi

Bay	No.	Container Priority	Time Spent (minutes)
1	1	1	6
1	2	2	6
1	3	3	6
1	4	4	6
1	5	5	6
1	6	6	6
2	7	1	6
2	8	2	6
2	9	3	6
2	10	4	6
2	11	5	6
2	12	6	6
3	13	1	6
3	14	2	6
3	15	3	6
3	16	4	6
3	17	5	6
3	18	6	6
4	19	1	6
4	20	2	6
4	21	3	6
4	22	4	6
4	23	5	6
4	24	6	6
5	25	1	6
5	26	2	6

(c) Waktu yang dibutuhkan untuk setiap pengambilan kontainer

## Lampiran 4. Model Simulasi



## Lampiran 5. Dokumentasi Kunjungan ke Terminal Peti Kemas New Makassar 1





## Lampiran 6. Daftar hadir dan berita acara seminar hasil



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

---

**DAFTAR HADIR SEMINAR HASIL**

Nama/Stambuk : I.Shekinah Queeny Limuang D121201056

Judul Skripsi/T.A : "Simulasi Pemodelan Sistem Reduksi Waktu Relokasi Kontainer Menggunakan Metode Heuristik ( Studi Kasus Terminal Petikemas New Makassar 1 )" "

Hari/Tanggal : Rabu, 10 Juli 2024

Jam : 13.00 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1..... 
	Pembimbing II	2. Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D	2..... 
II.	Anggota Penguji	3. Prof.Dr.Ir. Indrabayu,ST.,M.T.,M.Bus.IPM.ASEAN.Eng 3	3..... 
		4. Dr.Eng. Ir. Zulkifli Tahir,ST.,M.Sc	4..... 

---

**PANITIA UJIAN**

Ketua,

Dr. Ir. Ingrid Nurtanio, M.T

Sekretaris

Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**BERTA ACARA SEMINAR HASIL**

Pada hari ini Rabu, tanggal 10 Juli 2024 Pukul 13.00 WITA - Selesai bertempat di Ruang Lab. CBS Departemen Teknik Informatika, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Shekinah Queeny Limuang  
No. Stambuk : D121201056  
Fakultas/Departemen : Teknik/Teknik Informatika  
Judul Skripsi : "Simulasi Pemodelan Sistem Reduksi Waktu Relokasi Kontainer Menggunakan Metode Heuristik ( Studi Kasus Terminal Petikemas New Makassar I )"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1.
2.	Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D	Pemb II/Sekretaris	2.
3.	Prof.Dr.Ir. Indrabayu,ST.,M.T.,M.Bus.IPM.ASEAN.Eng	Anggota	3.
4.	Dr.Eng. Ir. Zulkifli Tahir,ST.,M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Seminar Hasil : Lulus / Tidak lulus dengan nilai angka .....<sup>87</sup> dan huruf ...<sup>A</sup>.....

Gowa, 10 Juli 2024

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Nomor : 900/UN4.7.7/TD.06/2024  
Lamp : -  
Hal : Penerbitan Surat Penugasan Panitia/Penguji  
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,

Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia/Penguji Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Shekinah Queeny Limuang D121201056  
Judul TA : Simulasi Pemodelan Sistem Reduksi Waktu Relokasi Kontainer  
Menggunakan Metode Heuristik (Studi Kasus Terminal Peti  
Kemas New Makassar 1).

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T  
Pembimbing II / Sekretaris : 2. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D  
Anggota : 3. Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys  
4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 8 Juli 2024

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys., IPM, ASEAN.Eng  
Nip.19750716 200212 1 004

Tembusan :  
1. Arsip

Rabu, 10 Juli 2024  
Jum, 13<sup>00</sup>  
Sab, CBS





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS HASANUDDIN**  
**FAKULTAS TEKNIK**

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan  
☎ +62811 4420 909, E-mail: [teknik@unhas.ac.id](mailto:teknik@unhas.ac.id), <https://eng.unhas.ac.id>

**SURAT PENUGASAN**  
No. 15736/UN4.7.1/TD.06/2024

- Dari : Dekan Fakultas Teknik Universitas Hasanuddin  
Kepada : Mereka yang tercantum namanya dibawah ini  
Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin Nomor : **29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin**, dengan ini menugaskan Saudara sebagai PENGUJI/PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :
- |                            |  |
|----------------------------|--|
| Pembimbing I/Ketua         | : 1. Dr. Ir. Ingrid Nurtanio, M.T                |
| Pembimbing II / Sekretaris | : 2. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D        |
| Anggota                    | : 3. Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys |
|                            | : 4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.      |

Untuk menguji bagi mahasiswa tersebut dibawah ini :

- |                      |   |            |
|----------------------|---|------------|
| Nama/NIM             | : Shekinah Queeny Limuang   | D121201056 |
| Program Studi        | : Teknik Informatika  |            |
| Judul thesis/Skripsi | : Simulasi Pemodelan Sistem Reduksi Waktu Relokasi Kontainer Menggunakan Metode Heuristik (Studi Kasus Terminal Peti Kemas New Makassar 1). |            |
- Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
  - Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
  - Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa  
Pada tanggal 8 Juli 2024  
a.n. Dekan,  
Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT  
NIP. 197310101998021001

- Tembusan :
- Dekan Fak. Teknik Unhas
  - Ketua Departemen Teknik Informatika FT-UH
  - Mahasiswa yang bersangkutan



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE  
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1  
\*Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah\*

## Lampiran 7. Daftar hadir dan berita acara ujian skripsi



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**DAFTAR HADIR UJIAN SKRIPSI MAHASISWA  
FAKULTAS TEKNIK UNHAS**

Nama/Stambuk : 1. Shekinah Queeny Limuang D121201056

Judul Skripsi/T.A : "Simulasi Pemodelan Sistem Relokasi Kontainer Menggunakan Algoritma Genetika ( Studi Kasus Terminal Peti kemas New Makassar 1 )"

Hari/Tanggal : Rabu, 14 Agustus 2024

Jam : 10- 30 Wita – Selesai

Tempat : Ruang Lab. AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1.
	Pembimbing II	2. Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D	2.
II.	Anggota Penguji 3.	3. Prof.Dr.Ir. Indrabayu,ST.,M.T.,M.Bus.IPM.ASEAN.Eng 3	3.
		4. Dr.Eng. Ir. Zulkifli Tahir,ST.,M.Sc	4.

PANITIA UJIAN

Ketua,  
  
Dr. Ir. Ingrid Nurtanio, M.T

Sekretaris  
  
Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://cng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**BERITA ACARA UJIAN SKRIPSI**

Pada hari ini Rabu, tanggal 14 Agustus 2024 Pukul 10.30 WITA - Selesai bertempat di Lab. AIMP Departemen Teknik Informatika Gowa, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Shekinah Queeny Limuang  
No. Stambuk : D121201056  
Fakultas/Departemen : Teknik /Teknik Informatika  
Judul Skripsi : "Simulasi Pemodelan Sistem Relokasi Kontainer Menggunakan Algoritma Genetika ( Studi Kasus Terminal Peti kemas New Makassar 1 )"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1.
2.	Mukarramah Yusuf, B.Sc.,M.Sc.,Ph.D	Pemb II/Sekretaris	2.
3.	Prof.Dr.Ir. Indrabayu,ST.,M.T.,M.Bus.IPM.ASEAN.Eng	Anggota	3.
4.	Dr.Eng. Ir. Zulkifli Tahir,ST.,M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : ~~Lulus~~ / Tidak lulus dengan nilai angka ..... 87 ..... dan huruf (A) .....

Gowa, 14 Agustus 2024

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA**  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Gowa, 9 Agustus 2024

Nomor : 1133/UN4.7.7.1/TD.06/2024  
Lamp : -  
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana  
  
Yth : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas  
Di  
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio. M.T.  
Pembimbing II / Sekretaris : 2. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.  
Anggota : 3. Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys.  
4. Dr.Eng. Zulkifli Tahir, S.T., M.Sc.

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Shekinah Queeny Limuang  
Stambuk : D121 20 1056

Dengan Judul Skripsi :

“ Simulasi Pemodelan Sistem Relokasi Kontainer Menggunakan Algoritma Genetika  
(Studi Kasus Terminal Peti Kemas New Makassar 1) “

Pada :  
Hari/Tanggal : Rabu, 14 Agustus 2024  
Jam : 10.30 Wita - Selesai  
Tempat : Ruang Sidang Lab. AIMP

Demikian penyampaian kami, atas perhatiannya diucapkan terimakasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng  
Nip.197507016 200212 1 004

Tembusan :  
1. Arsip





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan  
☎ +62811 4420 909, E-mail: [teknik@unhas.ac.id](mailto:teknik@unhas.ac.id), <https://eng.unhas.ac.id>

**SURAT PENUGASAN**

No. 19626/UN4.7.1/TD.06/2024

- Dari : Dekan Fakultas Teknik Universitas Hasanuddin.  
Kepada : Mereka yang tercantum namanya di bawah ini.  
Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin, dengan ini menugaskan Saudara sebagai PENGUJI/PANITIA UJIAN SARJANA Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :
- Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio. M.T.  
Pembimbing II / Sekretaris : 2. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.  
Anggota : 3. Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys.  
4. Dr.Eng. Zulkifli Tahir, S.T., M.Sc.

untuk menguji bagi mahasiswa tersebut di bawah ini :

- Nama/NIM : Shekinah Queeny Limuang D121201056  
Program Studi : Teknik Informatika  
Judul Thesis/Skripsi : Simulasi Pemodelan Sistem Relokasi Kontainer Menggunakan Algoritma Genetika (Studi Kasus Terminal Peti Kemas New Makassar 1).
- Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
  - Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
  - Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,  
Pada tanggal 12 Agustus 2024  
a.n. Dekan  
Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT  
NIP.197310101998021001

**Tembusan :**

- Dekan Fak. Teknik Unhas
- Ketua Departemen Teknik Informatika FT-UH
- Kasubag. Umum dan Perlengkapan FT-UH



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSRE  
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1  
\*Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah\*








## Lampiran 8. Lembar perbaikan skripsi

**LEMBAR PERBAIKAN SKRIPSI****“SIMULASI PEMODELAN SISTEM RELOKASI KONTAINER  
MENGUNAKAN ALGORITMA GENETIKA (STUDI KASUS  
TERMINAL PETI KEMAS NEW MAKASSAR 1)”****OLEH:****SHEKINAH QUEENY LIMUANG  
D121 20 1056**


Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 14 Agustus 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Ir. Ingrid Nurtanio, M.T.	
Sekretaris	Mukarramah Yusuf, B.Sc., M.Sc., Ph.D	
Anggota	Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM. ASEAN. Eng.	
	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T..	
II	Mukarramah Yusuf, B.Sc., M.Sc., Ph.D	