

Perancangan Sistem Optimasi Irigasi Di Manganti

Athar F - 8 August 2024
- V.0 : Initial Release

Daftar Isi

1.	Latar Belakang.....	1
2.	Perumusan Masalah.....	1
3.	Objektif.....	2
4.	Overview Sistem Utama.....	2
5.	Spesifikasi Input Output.....	2
	Input.....	2
	Output.....	3
6.	Pemilihan Solusi	3
7.	Perancangan Sistem.....	4
	Metode Recurrent Neural Network.....	4
	a. Pembahasan.....	4
	b. Tahapan Perancangan.....	4
	1. Data Collection and Preprocessing	5
	2. Model Architecture Design	6
	3. Model Training	6
	4. Model Deployment	7
	5. Model Validation and Fine-Tuning.....	7
	6. Continuous Monitoring and Updates	7

1. Latar Belakang

N/A

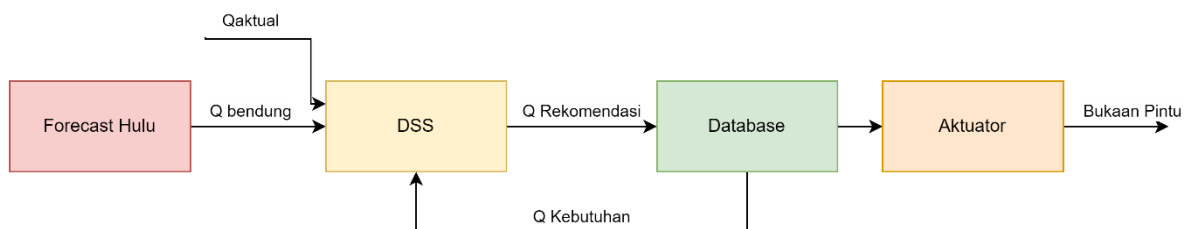
2. Perumusan Masalah

Menentukan pembagian irigasi paling optimasl dalam menghasilkan produktivitas tanam maksimal dengan waktu minimal

3. Objektif

Objektif	Deskripsi	Evaluasi
Alokasi Air	Pembagian debit air pada setiap petak tersier	<ul style="list-style-type: none">• Debit terdistribusi per kebutuhan• Indeks Pertanaman
Waktu	Waktu yang dibutuhkan dalam memnuhi kebutuhan setiap petak	<ul style="list-style-type: none">• Waktu tempuh• Waktu pemenuhan air

4. Overview Sistem Utama



Sistem Optimasi irigasi tergabung dalam bagian Decision Support System.

5. Spesifikasi Input Output

Input

- a. Ketersediaan Air di Bendung Manganti(Q_0)

Air yang tersedia di bendung Manganti yaitu histori, realtime, serta hasil forecast

- b. Kebutuhan air di Petak Tersier (Q_N)

Jumlah debit air yang dibutuhkan setiap petak tersier

- c. Skema Jaringan Irigasi

Skema irigasi DI Manganti yang menentukan aliran air dari bendung menuju ke setiap petak tersier

- d. Debit air di Petak tersier (Q_A)

Jumlah debit air aktual dan historis setiap petak tersier

e. Data Cuaca (Opsional)

Data faktor cuaca yang memengaruhi aliran air irigasi (curah hujan, evaporasi, dll)

f. Data faktor sosial (Opsional)

Faktor sosial berupa masukan dari lapangan terkait kesiapan tanam dll

Output

a. Rekomendasi alokasi air (Q_R)

Debit air yang direkomendasikan untuk setiap petak untuk jangka waktu tertentu

Contoh Output

	T1	T2	T3	T4	T5	T6	T7
Petak a	$Q_{Ra}(T1)$	$Q_{Ra}(T2)$	$Q_{Ra}(T3)$	$Q_{Ra}(T4)$	$Q_{Ra}(T5)$	$Q_{Ra}(T6)$	$Q_{Ra}(T7)$
Petak b	$Q_{Rb}(T1)$	$Q_{Rb}(T2)$	$Q_{Rb}(T3)$	$Q_{Rb}(T4)$	$Q_{Rb}(T5)$	$Q_{Rb}(T6)$	$Q_{Rb}(T7)$

Keterangan

TN : deskripsi waktu (hari atau jam)
Petak x : nama petak tersier
 $Q_{Rx}(TN)$: Rekomendasi debit untuk petak x pada waktu TN

6. Pemilihan Solusi

- Algoritma Optimasi :

Membangun model matematis jaringan irigasi untuk mendapatkan nilai optimal yaitu memaksimalkan produktivitas dan efektivitas, meminimalisir waktu

- Recurrent Neural Network :

Menggunakan data historis temporal yang diberi label untuk mendapatkan hubungan input output dari sistem

- Reinforcement Learning :

Metode trial dan error untuk mendapatkan model optimal pembagian air sistem jaringan irigasi

7. Perancangan Sistem

Metode Recurrent Neural Network

a. Pembahasan

Recurrent neural network (RNN) adalah jenis ANN yang digunakan untuk mengenali pola pada data sekuensial, seperti data waktu. RNN dilengkapi dengan sebuah “memori” input sebelumnya yang berpengaruh terhadap data pada sekuens berikutnya.

Dalam optimasi irigasi, RNN dapat digunakan untuk memprediksi dan mengatur distribusi air dalam periode waktu yang output bergantung kepada output sebelumnya dan kondisi saat ini.

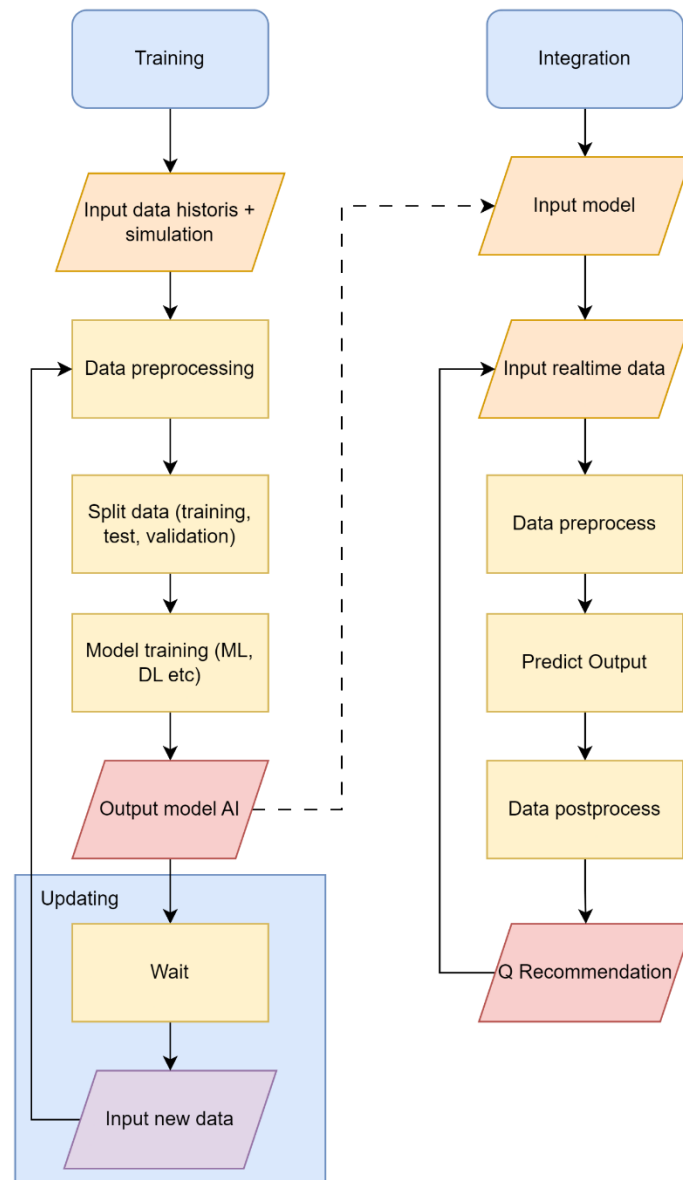
Keuntungan dalam penggunaan RNN

1. Menangkap pola berbasis waktu
2. Memprediksi dan merencanakan sekuens rekomendasi distribusi air untuk jangka waktu tertentu
3. Berdaptasi untuk input yang dinamis
4. Menangkap pola rotasi distribusi air
5. Mengintegrasikan berbagai macam faktor

Jenis-jenis RNN

1. Long Short Term Memory
2. Gated Recurrent Network

b. Tahapan Perancangan



1. Data Collection and Preprocessing

Tahap 1.1: Pengumpulan Data Historis.

- Pengumpulan data historis sesuai dengan spesifikasi input output

Step 1.2: Preprocess the Data

- **Normalisasi:** Normalisasi data untuk mendapatkan hasil yang baik.
- **Format data waktu:** Merapikan data menjadi sequences yang merepresentasikan setiap faktor per satuan waktu (kebutuhan, ketersediaan, dll.).
- **Penanganan data kosong:** Menggunakan interpolasi untuk mengisi kekosongan yang ada
- **Feature Engineering:** Membuat tambahan fitur untuk tren beberapa hari sebelumnya (contoh, trends ketersediaan).

2. Model Architecture Design

Step 2.1: Input Layer

- **Inputs:**
 - **Distribusi air sebelumnya:** Sekuens distribusi air untuk setiap petak sawah beberapa hari sebelumnya
 - **Data Presipitasi:** Sekuens data sebelumnya dan prediksi dari presipitasi untuk setiap sawah.
 - **Kebutuhan Air:** Kebutuhan air masukan dari pengguna
 - **Ketersediaan Air:** Air yang tersedia, aktual dan prediksi

Step 2.2: RNN Layers

- **LSTM/GRU Layers:**
 - Menggunakan 1 atau 2 layer LSTM/GRU untuk menangkap ketergantungan temporal pada seluruh urutan input.
- **Sequence-to-Sequence Architecture:**
 - Mengimplementasikan arsitektur sequence-to-sequence (Seq2Seq) di mana model mengambil urutan pengamatan masa lalu dan mengeluarkan urutan prediksi untuk 14 hari berikutnya.
 - **Encoder:** Menginput sekuens masukan (e.g., data 7 hari terakhir).
 - **Decoder:** Menghasilkan output data prediksi (e.g., prediksi distribusi air untuk 14 hari).

Step 2.3: Output Layer

- **Jenis Output:**
 - Sekuens prediksi distribusi air untuk setiap petak sawah selama jangka waktu tertentu (contoh, 14 hari ke depan).
- **Activation Function:** Menggunakan fungsi aktivasi linear untuk mendapatkan prediksi yang bersifat kontinyu.

3. Model Training

Step 3.1: Loss Function

- **Objektif:**
 - Meminimalisir galat antara distribusi air yang diprediksi dan aktual selama jangka waktu tertentu (14 hari).
 - Memberikan penalti jika berdasarkan kebutuhan yang tidak terpenuhi, melebihi ketersediaan, dan konstrain waktu.
 - **Loss Function** dirumuskan berdasarkan parameter tersebut

Step 3.2: Pemilihan algoritma optimisasi

- **Adam Optimizer:** Menggunakan optimisasi adam untuk training model.
- **Learning Rate:** Learning rate disesuaikan dan dicari nilai yang paling cocok.

Step 3.3: Proses Training

- **Batch Size and Epochs:** Sesuaikan hyperparameter tersebut untuk mendapatkan hasil yang terbaik
- **Validation Split:** Gunakan sebagian data (20%) sebagai validasi hasil training.
- **Early Stopping:** Implementasi early stopping untuk menghindari overfitting.

4. Model Deployment

Step 4.1: Prediksi Real-Time

- **Input Real-Time Data:** Masukkan input real-time termasuk prakiraan curah hujan terkini, kebutuhan air saat ini, dan supply air di bendung.
- **Membuat Prediksi:**
 - Model RNN menghasilkan prediksi rekomendasi distribusi air untuk setiap petak, mempertimbangkan data sebelumnya.

Step 4.2: Feedback Loop

- **Pembelajaran Berkelanjutan:** Perbarui model secara berkala dengan data baru dan perbaiki prediksi berdasarkan umpan balik dari hasil distribusi aktual.

5. Model Validation and Fine-Tuning

Step 5.1: Cross-Validation

- **Time Series Cross-Validation:** Metode validasi *sliding window* dapat digunakan untuk memastikan model menghasilkan hasil yang baik pada masa mendatang
- **Pengujian Skenario:** Evaluasi model untuk kondisi yang berbeda-beda.

Step 5.2: Evaluasi Performa

- **Akurasi:** Menilai seberapa akurat model memprediksi distribusi air selama periode waktu rekomendasi.
- **Efisiensi Air:** Mengevaluasi seberapa baik model mengelola persediaan air yang tersedia selama prakiraan waktu rekomendasi.
- **Efisiensi Waktu:** Pastikan distribusi air tepat waktu untuk memenuhi kebutuhan petak sawah dalam kurun waktu prediksi.

6. Continuous Monitoring and Updates

Step 6.1: Monitoring

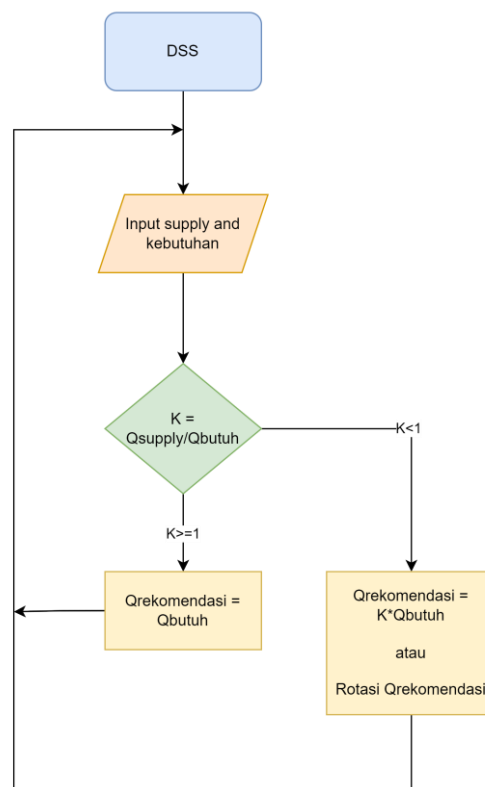
- **Performance Tracking:** Monitor akurasi dan efisiensi hasil output real-time.
- **Penyesuaian:** Membuat penyesuaian terhadap model atau training ulang jika dibutuhkan.

Step 6.2: Model Updates

- **Regular Updates:** Update berkala untuk memastikan akurasi terhadap perubahan di lapangan.

8. Implementasi Sistem

Implementasi Sistem Sementara (Staging 1)



Pada tahap ini, rekomendasi yang diberikan berdasarkan perhitungan nilai faktor K, dengan nilai $K \geq 1$ maka diberikan rekomendasi sesuai kebutuhan sedangkan jika $K < 1$, maka dilakukan pengurangan/rotasi.