

**VISUALISASI ADAPTIF KONDISI UHI BERBASIS PREDIKSI  
MENGUNAKAN *SUPPORT VECTOR REGRESSION***

**TESIS**

**Oleh:  
SALMA AINUR ROHMA  
NIM. 240605210016**



**PROGRAM STUDI MAGISTER INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**VISUALISASI ADAPTIF KONDISI UHI BERBASIS PREDIKSI  
MENGUNAKAN *SUPPORT VECTOR REGRESSION***

**TESIS**

**Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:  
SALMA AINUR ROHMA  
NIM. 240605210016**

**PROGRAM STUDI MAGISTER INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**VISUALISASI ADAPTIF KONDISI UHI BERBASIS PREDIKSI  
MENGUNAKAN *SUPPORT VECTOR REGRESSION***

**THESIS**

**Oleh:**  
**SALMA AINUR ROHMA**  
**NIM. 240605110016**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 23 Oktober 2025

Pembimbing I,



Dr. Ir. Yunifa Miftachul Arif, M.T  
NIP 19830616 201101 1 004

Pembimbing II,



Dr. Tarranita Kusumadewi, M.T  
NIP. 19790913 200604 2 001

Mengetahui,  
Ketua Program Studi Magister Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Prof. Dr. Muhammad Faisal, S. Kom., M.T.  
NIP. 19740510 200501 1 007

**VISUALISASI ADAPTIF KONDISI UHI BERBASIS PREDIKSI  
MENGUNAKAN SUPPORT VECTOR REGRESSION**

**THESIS**

**Oleh:  
SALMA AINUR ROHMA  
NIM. 240605110016**

Telah Dipertahankan di Depan Dewan Penguji Thesis  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Magister Komputer (M.Kom)  
Tanggal: 23 Oktober 2025

**Susunan Dewan Penguji**

Penguji I	: <u>Dr. Ririen Kusumawati, S.Si, M.Kom</u> NIP. 19720309 200501 2 002
Penguji II	: <u>Dr. Ir. Fresy Nugroho, M.T, IPM</u> NIP. 19710722 201101 1 001
Pembimbing I	: <u>Dr. Ir. Yunifa Miftachul Arif, M.T</u> NIP 19830616 201101 1 004
Pembimbing II	: <u>Dr. Tarranita Kusumadewi, M.T</u> NIP. 19790913 200604 2 001

()  
()  
()  
()

Mengetahui dan Mengesahkan,  
Ketua Program Studi Magister Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Prof. Dr. Ir. Muhammad Faisal, S. Kom., M.T.  
NIP. 19740510 200501 1 007

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : SALMA AINUR ROHMA  
NIM : 240605210016  
Fakultas / Prodi : Sains dan Teknologi / Magister Informatika  
Judul Thesis : VISUALISASI ADAPTIF KONDISI UHI BERBASIS REDIKSI  
MENGGUNAKAN *SUPPORT VECTOR REGRESSION*

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Oktober 2025  
Yang membuat pernyataan,



SALMA AINUR ROHMA  
NIM. 240605210016

MOTTO

وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ

*“Teruslah berbuat baik, sebagaimana Allah selalu menyiapkan yang terbaik untukmu”*

## **HALAMAN PERSEMBAHAN**

Segala puji bagi Allah SWT yang telah memberikan limpahan rahmat dan hidayah kepada hamba-Nya, sehingga saya bisa menyelesaikan karya ini. Serta shalawat dan salam semoga selalu tercurahkan kepada nabi Muhammad SAW.

Saya persembahkan karya ini kepada:

Ayah dan ibu saya,

Suratno, S.Pd., dan Anita Mustikasari, S.Pd.

Terima kasih atas segala ilmu, dukungan dan semangatnya serta yang tidak kalah penting adalah doa di setiap pagi dan malamnya, sehingga saya bisa menyelesaikan ujian demi ujian khususnya skripsi ini. semoga karya ini menjadi karya yang bermanfaat dunia dan akhirat serta menjadi keberkahan untuk semuanya.

Teruntuk kakak dan saudara kembar saya,

Viranita Quratul Aini, S.Si dan Shima Ainur Rofi, S.Psi.

Terima kasih atas semangat, dukungan, kesabaran, dan doa serta selalu ada baik saat senang maupun duka

Tim Penelitian serta pihak yang membantu penelitian

Terima kasih atas ilmu, pengalaman, dan kesempatannya serta dukungan yang selalu diberikan. Sehingga saya bisa menyelesaikan karya ini.

Teman-teman seperjuangan,

Seruluh Mahasiswa Teknik Informatika khususnya untuk Angkatan 2020 yang tidak bisa saya sebutkan satu-persatu

Yang telah berbagi ilmu, support serta pengalaman yang berharga selama perkuliahan ini. Semoga kita semua selalu diberi kesehatan dan kemudahan untuk menuntut ilmu serta dimudahkan dalam urusan lainnya oleh Allah SWT.

## KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarakatuh.*

Segala puji hanya milik Allah Subhanahu Wa Ta'ala atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan skripsi yang berjudul “Visualisasi Adaptif Kondisi Lingkungan Virtual untuk Simulasi UHI Kota Malang Menggunakan *Support Vector Regression*”. Semoga shalawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallallahu ‘Alaihi wa Sallam. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih yang tak terhingga kepada semua pihak-pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk dapat menyelesaikan skripsi ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Muhammad Faisal, MT, selaku Ketua Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Yunifa Miftachul Arif, M.T selaku dosen pembimbing I dan Dr. Tarranita Kusumadewi, M.T. selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan thesis ini.
5. Dr. Ririen Kusumawati, M.Kom selaku dosen penguji I dan Dr. Fresy Nugroho, M.T selaku dosen penguji II yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan thesis dengan baik.



6. Segenap Dosen, Admin, Laboran dan Mahasiswa Program Studi Magister Informatika yang telah memberikan banyak dukungan dan bimbingan selama pengerjaan skripsi ini.
7. Ibu, Ayah, Kakak, serta saudari kembar saya yang selalu memberikan dukungan dan motivasi untuk terus berusaha, dan doa yang tak putus-putusnya selalu disampaikan agar dapat menuntaskan skripsi ini dengan lancar dan baik.

Akhir kata, penulis mengakui bahwa penulisan pada skripsi ini masih banyak kekurangan. Saya berharap semoga skripsi ini diterima sebagai amal ibadah yang tulus dan bermanfaat di sisi Allah Subhanahu Wa Ta'ala. Semoga karya ini menjadi bagian dari kontribusi yang tak terputus dalam rangka memperkuat dan mengembangkan ilmu pengetahuan, serta melaksanakan tugas sebagai hamba Allah yang berkomitmen.

*Wassalamualaikum Warahmatullahi Wabarakatuh.*

Malang, 21 Oktober 2025

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PENGANTAR .....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>v</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR .....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiv</b>
<b>ABSTRAK .....</b>	<b>xv</b>
<b>ABSTRACT .....</b>	<b>xvi</b>
<b>مستخلص البحث .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Pernyataan Masalah .....	10
1.3 Tujuan Penelitian .....	10
1.4 Batasan Masalah .....	10
1.5 Manfaat Penelitian .....	11
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>12</b>
2.1 Fenomena UHI .....	12
2.2 Faktor-faktor yang mempengaruhi UHI .....	13
2.3 Machine Learning .....	16
2.4 SVR .....	19
2.5 Visualisasi Berbasis Machine Learning .....	22
2.6 Visualisasi UHI .....	24
2.7 Visualisasi Adaptif pada Ruang Virtual .....	25
2.8 Integrasi API untuk Penghubung Data Nyata, Model Prediktif, dan Visualisasi Adaptif .....	27
2.9 Evaluasi Performa .....	30
2.10 Kerangka Teori .....	35
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>39</b>
3.1 Study Area .....	39
3.2 Desain Penelitian .....	41
3.2.1 Pengumpulan Data .....	44
1. Landsat-8 .....	46
2. Sentinel-2 Level-2A .....	47
3. SRTM (Shuttle Radar Topography Mission) .....	53
3.2.2 Pengembangan Model Prediksi .....	55
3.2.2.1 Normalisasi Data .....	55
3.2.2.2 Pembagian Data .....	56
3.2.2.3 Metode SVR .....	58

3.2.2.4	Pemilihan Kernel dan Tuning Hyperparameter .....	59
3.2.2.5	Pembentukan Kernel, Matriks Hessian, dan Proses Sequential Learning .....	60
3.2.2.6	Evaluasi Performa Model.....	60
1.	Koefisien determinasi ( $R^2$ ).....	60
2.	Mean Absolute Percentage <i>Error</i> (MAPE) .....	61
3.2.2.7	Skenario Pengujian dan Pemilihan Model Terbaik.....	62
3.2.3	Tahap Integrasi API.....	62
3.2.4	Visualisasi Adaptif dalam Lingkungan VR .....	63
3.2.4.1.	Desain Sistem.....	64
<b>BAB IV PERANCANGAN DAN PENGEMBANGAN 3D CITY MODEL .....</b>		<b>84</b>
4.1	Pembuatan 3D City Model .....	85
4.2	Penentuan Titik Koordinat .....	94
4.3	Integrasi Model dan Data ke Unity .....	97
<b>BAB V DATA REMOTE SENSING DAN MODEL SVR .....</b>		<b>101</b>
5.1	Pengumpulan Data dan Preprocessing .....	101
5.1.1	Cloud Masking.....	101
5.1.2	Feature Scalling .....	103
5.2	Normalisasi Data .....	103
5.3	<i>Data Splitting</i> .....	104
5.4	Hasil Pengujian Model SVR .....	106
5.4.1	Hasil pengujian Skenario 1.....	106
5.4.2	Hasil pengujian Skenario 2.....	110
5.4.3	Hasil pengujian Skenario 3.....	114
5.5	Analisis Perbandingan Antar Skenario .....	117
5.6	Hasil Prediksi Model SVR.....	121
<b>BAB VI ANALISIS EVALUASI PERFORMA MODEL .....</b>		<b>123</b>
6.1	Analisis Pemilihan Kernel RBF sebagai Model Terbaik.....	123
6.2.1	Perbandingan Kinerja Antar Kernel .....	123
6.2.2	Stabilitas dan Kemampuan Generalisasi Model .....	125
6.2	Analisis Pengaruh Hyperparameter terhadap Performa Model.....	126
6.2.1	Parameter Cost (C) .....	126
6.2.2	Parameter Epsilon ( $\epsilon$ ) .....	126
6.2.3	Parameter Gamma ( $\gamma$ ) .....	127
6.3	Analisis Kestabilan dan Robustness Model .....	128
6.4	Implikasi terhadap Visualisasi UHI di Game VR.....	129
<b>BAB VII PERANCANGAN DAN INTEGRASI MODEL SVR DENGAN VISUALISASI ADAPTIF BERBASIS API.....</b>		<b>131</b>
7.1	Integrasi Model Prediktif dengan API Backend.....	131
7.2	Konversi dan Penyelarasan Data Prediksi untuk Unity .....	133

7.3	Arsitektur API Backend .....	135
7.4	Mekanisme Komunikasi Unity dan API .....	138
7.5	Integrasi Unity sebagai Client Visualisasi .....	139
7.6	Penanganan <i>Error</i> , Caching, Monitoring, dan Skalabilitas .....	141
7.7	Evaluasi Performa Integrasi Sistem .....	142
7.7.3	Efisiensi dan Optimasi Data .....	145
7.7.4	Validasi Interkoneksi dan Integrasi End-to-End .....	146
<b>BAB VIII IMPLEMENTASI VISUALISASI ADAPTIF DALAM LINGKUNGAN VR.....</b>		<b>148</b>
8.1	Arsitektur Sistem Visualisasi Adaptif .....	149
8.2	Integrasi Data ke Dalam Unity 3D .....	151
8.3	Identifikasi Hotspot dan Penyajian Informasi Visual .....	155
8.4	Mekanisme Interaksi dan Navigasi Pengguna .....	156
<b>BAB IX EVALUASI <i>USABILITY</i> VISUALISASI ADAPTIF VR MENGGUNAKAN SUS .....</b>		<b>169</b>
9.1	Instrumen Penelitian .....	170
9.2	Responden dan Pengumpulan Data .....	171
9.3	Analisis Data SUS .....	174
<b>BAB X PEMBAHASAN .....</b>		<b>179</b>
<b>BAB XI PENUTUP .....</b>		<b>188</b>
11.1	Kesimpulan .....	188
11.2	Saran .....	189
<b>LAMPIRAN.....</b>		<b>193</b>
Lampiran 1 Data Non-Linier .....		193
Lampiran 2 Psudocode unity web request ke server .....		196
Lampiran 3 Psudocode Data Input, Validasi, dan Persiapan Data GEE .....		197
Lampiran 4 Psudocode Pemrosesan GEE, Prediksi Model, dan Output .....		198
Lampiran 5 Psudocode koordinat unity .....		199

## DAFTAR TABEL

Tabel 2. 1 Faktor yang mempengaruhi UHI .....	13
Tabel 2. 2 Evaluasi Performa Sistem Visualisasi Adaptif .....	33
Tabel 3. 1 Citra Satelit yang akan digunakan .....	46
Tabel 3. 2 Fitur-fitur yang akan digunakan.....	48
Tabel 3. 3 Kriteria Nilai MAPE (F. Zhang & O'Donnell, 2019).....	61
Tabel 3. 4 Klasifikasi kelas UHI (Nainggolan <i>et al.</i> , 2020).....	73
Tabel 3. 5 Variabel Skala Likert (Budiaji, 2013).....	82
Tabel 5. 1 Feature Calculation .....	103
Tabel 5. 2 Sebelum dan Sesudah Normalisasi data.....	104
Tabel 5. 3 Rasio Splitting Data .....	105
Tabel 5. 4 Perbandingan hasil pengujian terbaik pada setiap skenario.....	118
Tabel 7. 1 Proses waktu eksekusi.....	143
Tabel 9. 1 Instrumen Pertanyaan SUS .....	170

## DAFTAR GAMBAR

Gambar 2. 1 Kerangka Teori.....	36
Gambar 4. 1 Tahapan umum perancangan dan pengembangan 3D.....	85
Gambar 4. 2 Tampilan model tiga dimensi Kota Malang di Blender menggunakan plugin BIOSM .....	86
Gambar 4. 3 Generate Elevation terain menggunakan SRTM.....	87
Gambar 4. 4 Penerapan tekstur homogen pada terrain .....	90
Gambar 4. 5 Penerapan LOD .....	91
Gambar 4. 6 Distribusi titik koordinat representatif Kota Malang menggunakan plugin <i>Regular Points</i> di QGIS. ....	95
Gambar 4. 7 Integrasi model 3D Kota Malang ke dalam Unity 3D untuk sistem visualisasi UHI berbasis VR. ....	98
Gambar 5. 1 Perbandingan visualisasi citra vegetasi menggunakan Sentinel-2 sebelum dan setelah dilakukan Cloud Masking .....	102
Gambar 5. 2 $R^2$ Skenario 1 .....	108
Gambar 5. 3 MAPE Skenario 1.....	108
Gambar 5. 4 $R^2$ Skenario 2.....	111
Gambar 5. 5 MAPE Skenario 3.....	111
Gambar 5. 6 $R^2$ Skenario 3 .....	115
Gambar 5. 7 MAPE Skenario 3.....	115
Gambar 6. 1Peta Hasil Prediksi .....	128
Gambar 8. 1 Visualisasi Marker untuk hasil prediksi LST .....	152
Gambar 8. 2 Perbandingan efek LOD pada rendering marker dan objek hotspot untuk menjaga performa VR.....	153
Gambar 8. 3 <i>Heatmap</i> variabel LST .....	154
Gambar 8. 4 Titik Hotspot .....	155
Gambar 8. 5 Landing Page.....	156
Gambar 8. 6 Tutorial Dan Introduction Terkait UHI.....	157
Gambar 8. 7 Fetching data succes .....	158
Gambar 8. 8 Exploration State .....	159
Gambar 8. 9 ground mode.....	160
Gambar 8. 10 Fly Mode .....	160
Gambar 8. 11 Monitoring Hotspot .....	161
Gambar 8. 12 Titik H1 .....	162
Gambar 8. 13 Apabila jawaban bernilai True .....	163
Gambar 8. 14 Titik Hotspot 2 .....	164
Gambar 8. 15 Apabila Jawaban bernial False.....	165
Gambar 8. 16 Titik H3 .....	166
Gambar 9. 1 Rata-rata nilai SUS.....	176
Gambar 9. 2 Perbandingan Nilai SUS antar Kelompok Responden .....	177

## ABSTRAK

Rohma, Salma Ainur. 2025. **Visualisasi Adaptif Kondisi Lingkungan Virtual untuk Simulasi UHI Kota Malang Menggunakan Support Vector Regression**. Thesis. Jurusan Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Yunifa Miftachul Arif, M.T. (II) Dr. Tarranita Kusumadewi, M.T.

**Kata kunci:** *Urban Heat Island, Support Vector Regression, Virtual Reality, FastAPI, Usability.*

Penelitian ini bertujuan untuk mengembangkan sistem visualisasi adaptif berbasis VR yang mampu merepresentasikan fenomena UHI di Kota Malang secara interaktif. Sistem yang dikembangkan bernama Monitoring Malang City, mengintegrasikan model prediksi SVR dengan FastAPI dan Unity 3D, sehingga memungkinkan pemutakhiran data suhu permukaan tanah secara *real-time* melalui koneksi *API*. Arsitektur sistem dirancang dengan dua lapisan utama, yaitu *backend* untuk prediksi dan pemrosesan data dan *frontend* sebagai visualisasi termal adaptif. Model SVR dengan kernel RBF menghasilkan performa terbaik dengan nilai MAPE 2,76% dan  $R^2$  0,879, menunjukkan akurasi tinggi dalam estimasi *LST*. Hasil prediksi ini divisualisasikan ke dalam lingkungan 3D menggunakan sistem pewarnaan *heatmap* dan pengendalian interaksi berbasis *FSM*. Evaluasi *usability* menggunakan metode SUS terhadap 50 responden yang terdiri dari Bappeda, ahli game, dan mahasiswa, menunjukkan rata-rata skor 81,80, yang termasuk dalam kategori Good (Grade B). Hal ini menandakan bahwa sistem mudah digunakan, efisien, dan memberikan pengalaman interaktif yang memuaskan. Penelitian ini membuktikan bahwa integrasi SVR, API, dan VR dapat menghasilkan sistem simulasi cerdas yang tidak hanya menyajikan akurasi prediksi tinggi, tetapi juga meningkatkan pemahaman spasial pengguna terhadap dinamika termal perkotaan. Pendekatan ini diharapkan dapat berkontribusi pada pengembangan *smart city* dan pengelolaan lingkungan berbasis data ilmiah.

## ABSTRACT

Rohma, Salma Ainur. 2025. **Adaptive Visualization of Virtual Environmental Conditions for UHI Simulation of Malang City Using Support Vector Regression**. Thesis. Master Program of Computer Science, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Yunifa Miftachul Arif, M.T. (II) Dr. Tarranita Kusumadewi, M.T.

This study aims to develop an adaptive visualization system based on VR to represent the UHI phenomenon in Malang City interactively. The developed system, named Monitoring Malang City, integrates a SVR model with FastAPI and Unity 3D, allowing *real-time* updates of land surface temperature data through an API connection. The system architecture consists of two main layers: a *backend* for prediction and data processing, and a *frontend* for adaptive thermal visualization. The SVR model with an RBF kernel achieved the best performance with MAPE of 2.76% and  $R^2$  of 0.879, indicating high accuracy in *LST* estimation. The predicted data were visualized in a 3D environment using a thermal *heatmap* and a user interaction system controlled by a *FSM*. The *usability* evaluation, conducted using SUS with 50 respondents from *urban* planners, game experts, and university students, obtained an average score of 81.80, classified as Good (Grade B). This indicates that the system is user-friendly, efficient, and provides a satisfying interactive experience. The integration of SVR, API, and VR demonstrates the potential to produce an intelligent simulation system that not only achieves high predictive accuracy but also enhances users' spatial understanding of *urban* thermal dynamics. This approach contributes to the advancement of *smart city* development and data-driven environmental management.

**Keywords:** *Urban Heat Island, Support Vector Regression, Virtual Reality, FastAPI, Usability.*



## مستخلص البحث

روهما، سلمى أينور. ٢٠٢٥. التصوير التكميلي للظروف البيئية الافتراضية لمحاكاة ظاهرة الجزر الحرارية الحضرية في مدينة مالانج باستخدام خوارزمية دعم المتجهات (SVR). أطروحة ماجستير، برنامج ماجستير علوم الحاسب الآلي، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفان: (١) الدكتور يونيفا مفتاح العارف، ماجستير في التكنولوجيا. (٢) الدكتورة ترانيتا كسوما ديوي، ماجستير في التكنولوجيا.

**الكلمات المفتاحية:** الجزر الحرارية الحضرية، خوارزمية دعم المتجهات، الواقع الافتراضي، FastAPI، سهولة الاستخدام.

يهدف هذا البحث إلى تطوير نظام تصوير تفاعلي قائم على الواقع الافتراضي (VR) لتمثيل ظاهرة الجزر الحرارية الحضرية (UHI) في مدينة مالانج بطريقة تفاعلية. تم تطوير النظام المسمى Monitoring Malang City بدمج نموذج *Support Vector Regression (SVR)* مع FastAPI و Unity 3D، مما يتيح تحديث بيانات درجة حرارة سطح الأرض في الوقت الفعلي من خلال واجهة برمجة التطبيقات (API). يتكون هيكل النظام من طبقتين رئيسيتين: طبقة خلفية (backend) للتنبؤ ومعالجة البيانات، وطبقة أمامية (frontend) للتصوير الحراري التكميلي. أظهر نموذج SVR باستخدام نواة RBF أفضل أداء بقيمة MAPE تبلغ 2.76% ومعامل تحديد  $R^2$  يساوي 0.879، مما يشير إلى دقة عالية في تقدير درجة حرارة سطح الأرض. تم عرض النتائج في بيئة ثلاثية الأبعاد باستخدام خريطة حرارية (*heatmap*) ونظام تفاعل للمستخدم يعتمد على آلة الحالات المحدودة (FSM). أظهرت نتائج تقييم سهولة الاستخدام باستخدام مقياس *System Usability Scale (SUS)* بمشاركة 50 مشاركاً (من هيئة التخطيط العمراني، وخبراء الألعاب، والطلاب الجامعيين) متوسط درجة 81.80، مصنفاً ضمن الفئة جيد (الدرجة ب)، مما يدل على أن النظام سهل الاستخدام وفعال ويوفر تجربة تفاعلية مرضية. تؤكد نتائج هذا البحث أن دمج SVR و API و VR يمكن أن ينتج نظام محاكاة ذكي يتمتع بدقة تنبؤية عالية ويعزز فهم المستخدمين المكاني للديناميكيات الحرارية الحضرية. ويساهم هذا النهج في تطوير المدن الذكية وإدارة البيئة القائمة على البيانات العلمية. 48.22%.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan *urbanisasi* global yang pesat telah mengubah wajah kota-kota di seluruh dunia, termasuk di Indonesia. Kota kini tidak hanya menjadi pusat aktivitas ekonomi dan sosial, tetapi juga laboratorium nyata untuk pengembangan teknologi dan inovasi dalam perencanaan kota (Furuya *et al.*, 2023; Sun *et al.*, 2019). Transformasi ini membawa serta berbagai tantangan, terutama dalam pengelolaan lingkungan dan dampak ekologis yang ditimbulkan oleh aktivitas manusia. Salah satu dampak signifikan adalah fenomena *Urban Heat Island* (UHI), di mana suhu di area perkotaan meningkat akibat penggantian lahan alami dengan permukaan buatan seperti aspal dan beton (Santamouris, 2020).

Fenomena UHI telah menjadi perhatian global karena dampaknya yang luas, mulai dari penurunan kualitas udara, peningkatan konsumsi energi, hingga risiko kesehatan bagi penduduk kota. Perubahan penggunaan lahan dan peningkatan aktivitas antropogenik menyebabkan berkurangnya area hijau yang berperan sebagai penyerap panas alami, sehingga menimbulkan ketidakseimbangan termal di lingkungan perkotaan (Ren *et al.*, 2021). Di sisi lain, pertumbuhan populasi dan perkembangan infrastruktur kota yang terus meningkat mendorong kebutuhan akan solusi inovatif untuk memantau dan mengatasi permasalahan tersebut.

Dalam perspektif islam, tanggung jawab manusia terhadap lingkungan merupakan bagian dari amanah yang harus dijaga. Allah SWT berfirman dalam Surah Ar-Rum ayat 41:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

*“Telah tampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, Allah menghendaki agar mereka merasakan sebagian dari (akibat) perbuatan mereka agar mereka kembali (ke jalan yang benar)” (QS. Ar-Rum : 41).*

Menurut *Zubdatut Tafsir Min Fathil Qadir*, البحر merupakan daerah pemukiman yang berada di atas laut atau sungai. Sedangkan البر adalah daerah pemukiman yang berada di bawah laut atau sungai. Allah menjelaskan bahwa kemusyrikan dan kemaksiatan adalah sebab timbulnya kerusakan di alam semesta. Kerusakan ini dapat berupa kekeringan, paceklik, ketakutan yang merajalela, barang-barang yang tidak laku, sulitnya mencari penghidupan, maraknya perampokan dan kezaliman, dan lain sebagainya. Yakni agar mereka merasakan akibat dari sebagian perbuatan mereka sehingga menjauhi kemaksiatan mereka dan bertaubat kepada Allah.

Berdasarkan tafsir di atas dan melihat fenomena yang ada, suhu yang tinggi di area permukaan tanah dapat memberikan dampak buruk bagi kesehatan individu, seperti suhu panas, dehidrasi, dan gangguan pernapasan. Sebagai contoh, selama musim panas 2003 di Eropa, lebih dari 70.000 kematian disebabkan oleh gelombang panas. Tidak hanya gelombang panas, banyak faktor yang memengaruhi UHI, termasuk bentuk perkotaan, perubahan LULC (*Land Use and Land Cover*), perubahan iklim, dan emisi panas antropogenik (Liu *et al.*, 2021). Di antara faktor-faktor tersebut, perubahan LULC secara langsung berhubungan

dengan proses ekologi, seperti penyerapan radiasi matahari di permukaan dan evapotranspirasi sehingga memengaruhi perpindahan panas di permukaan dan menentukan intensitas UHI (Fitria *et al.*, 2019). Variasi pola lanskap mencerminkan bagaimana manusia mengubah penggunaan lahan (Liu *et al.*, 2021). Salah satu contoh nyata lainnya, yaitu di Kota Malang. Data BPS tahun 2021 menyebutkan pertumbuhan penduduk di Kota Malang mencapai 0,69% per tahun dengan tingkat kepadatan 9.582 jiwa/Km<sup>2</sup>. Dan dalam kurun waktu 5 tahun terakhir (2015-2020), Kota Malang teridentifikasi mengalami peningkatan intensitas UHI dengan semakin naiknya suhu permukaan tanah yang mencapai 1.7C (Kusumadewi *et al.*, 2023). Akibatnya suhu di Kota Malang mengalami kenaikan yang signifikan dan menyebabkan menurunnya kenyamanan termal di Kota Malang.

Dalam era *smart city*, integrasi teknologi informasi dan komunikasi telah membuka peluang baru dalam pengelolaan kota. Konsep *smart city* menekankan pentingnya pemanfaatan data *real-time*, sensor IoT, dan Sistem Informasi Geografis (SIG) untuk mengoptimalkan pengambilan keputusan dalam manajemen lingkungan dan perencanaan kota (Kamali Maskooni *et al.*, 2021). Salah satu pendekatan yang semakin berkembang adalah penggunaan visualisasi adaptif dalam lingkungan virtual, di mana data spasial dan temporal dapat ditampilkan secara interaktif dan dinamis. Visualisasi ini tidak hanya membantu dalam mendeteksi pola-pola lingkungan, tetapi juga memungkinkan pemodelan simulasi yang lebih akurat untuk mengantisipasi perubahan kondisi iklim lokal.

Penggunaan data penginderaan jauh, seperti citra satelit, telah terbukti efektif dalam mendokumentasikan kondisi termal permukaan kota (Dede *et al.*, 2020). Data ini memberikan gambaran rinci mengenai distribusi suhu di berbagai zona perkotaan, sehingga menjadi dasar penting dalam analisis UHI (Mathew *et al.*, 2019). Melalui pemrosesan dan analisis data tersebut, para peneliti dapat mengidentifikasi area-area yang paling rentan terhadap peningkatan suhu, serta merancang strategi mitigasi yang tepat. Namun, kompleksitas hubungan antara variabel lingkungan, seperti vegetasi, bangunan, kelembapan serta suhu, menuntut penggunaan metode analisis yang mampu menangani non-linearitas dan interaksi antar variabel.

Pemodelan suhu permukaan dalam studi UHI memerlukan pendekatan yang mampu menangkap hubungan kompleks antara berbagai variabel lingkungan, seperti tutupan lahan, kepadatan bangunan, albedo permukaan, vegetasi, dan faktor meteorologis. Sejumlah metode telah digunakan untuk tugas ini, seperti *Regresi Linier*, *Multiple Linear Regression* (MLR), *Decision tree*, ANN, dan *Support Vector Regression* (SVR). Namun, efektivitas setiap metode sangat bergantung pada karakteristik data dan kompleksitas hubungan antar variabel yang dianalisis. Oleh karena itu, diperlukan metode yang tidak hanya mampu menangkap hubungan non-linear, tetapi juga memiliki mekanisme pengendalian kompleksitas model agar tidak mudah mengalami *overfitting*. Dalam konteks ini, SVR menjadi salah satu metode yang paling relevan, karena SVR dirancang untuk mencari fungsi pemetaan terbaik dengan meminimalkan *error* dan memaksimalkan margin generalisasi. SVR juga memiliki kemampuan untuk

bekerja secara efektif pada data berukuran kecil hingga sedang, yang umum dijumpai dalam studi UHI berbasis citra satelit, sehingga menjadikannya kandidat metode regresi yang lebih adaptif dan stabil dibandingkan pendekatan konvensional.

Penelitian oleh Weng et al. (2004) menggunakan metode regresi linier untuk memodelkan suhu permukaan berdasarkan data penginderaan jauh, namun mereka mengakui keterbatasannya dalam menangkap hubungan non-linear, terutama ketika variabel-variabel spasial dan temporal sangat kompleks. Hal serupa diungkapkan dalam studi Myint et al. (2010), yang menyatakan bahwa regresi linier dan model statistik konvensional kurang fleksibel dalam menangani *noise* data serta pola hubungan yang tidak linier antara indeks vegetasi dan suhu permukaan. Sebagai respons terhadap keterbatasan tersebut, metode berbasis *Artificial Intelligent* seperti ANN mulai banyak digunakan. Misalnya, Zhou et al. (2018) memanfaatkan ANN untuk memodelkan distribusi suhu UHI dan memperoleh hasil yang cukup akurat. Namun, tantangan utama dari ANN adalah kebutuhan data pelatihan yang besar dan risiko *overfitting* yang tinggi, khususnya pada dataset spasial yang terbatas seperti di kota menengah.

Dalam studi yang lebih maju, SVR terbukti sebagai salah satu metode paling efisien dalam menangani prediksi suhu permukaan. Penelitian oleh Hsu, Chang, dan Lin (2003) menunjukkan bahwa SVR lebih tangguh terhadap *outlier* dan *noise* serta lebih mampu memodelkan hubungan non-linear antara input dan output dibanding regresi linier dan ANN. SVR juga memiliki keunggulan dalam regularisasi, yang menjadikannya ideal untuk dataset berukuran kecil hingga

sedang, seperti data suhu yang diperoleh dari titik pengamatan di wilayah perkotaan. Chen et al. (2020) menggunakan SVR untuk memprediksi *Land Surface Temperature* (LST) di wilayah *urban* Shanghai dan melaporkan bahwa SVR menghasilkan akurasi prediksi yang lebih tinggi dibandingkan *Random Forest* dan *Decision tree*. Hal ini diperkuat oleh studi Kumar et al. (2021), yang membandingkan SVR dengan beberapa algoritma regresi lainnya dalam konteks UHI di wilayah tropis dan menemukan bahwa SVR memberikan nilai RMSE dan MAE yang paling rendah, menjadikannya metode yang lebih andal dalam prediksi suhu. Konsistensi hasil dari berbagai studi tersebut menunjukkan bahwa SVR bukan hanya alternatif dari metode regresi tradisional, tetapi telah terbukti sebagai pendekatan yang lebih unggul dalam memodelkan variasi suhu permukaan, khususnya ketika variabel lingkungan memiliki hubungan yang kompleks dan non-linier. Hal ini memperkuat justifikasi pemilihan SVR dalam penelitian ini sebagai metode utama untuk memprediksi fenomena UHI secara akurat.

Khususnya untuk wilayah dengan karakteristik topografi dan penggunaan lahan yang kompleks seperti Kota Malang, kemampuan SVR dalam memodelkan ketidakaturan data menjadi sangat relevan. SVR dapat memanfaatkan *kernel functions* untuk memetakan hubungan non-linear ke dalam ruang dimensi tinggi, sehingga meningkatkan akurasi dalam prediksi suhu berdasarkan parameter spasial seperti tutupan lahan, indeks vegetasi, dan kerapatan bangunan. Selain itu, data penginderaan jauh yang digunakan dalam penelitian ini memiliki jumlah sampel yang tidak terlalu besar dan cenderung mengandung *noise* akibat variasi spasial dan temporal. SVR memiliki mekanisme regularisasi yang kuat untuk

mengontrol kompleksitas model, sehingga mampu menjaga keseimbangan antara bias dan varians. Dibandingkan dengan ANN yang membutuhkan data pelatihan besar atau *decision tree* yang rentan *overfitting*, SVR lebih stabil dalam kondisi data seperti ini. Dengan demikian, SVR tidak hanya dipilih karena keunggulan teoretisnya, tetapi juga karena kesesuaiannya dengan karakteristik data UHI di Kota Malang.

Integrasi antara visualisasi adaptif dalam lingkungan virtual dan algoritma SVR menawarkan pendekatan holistik dalam memodelkan fenomena UHI. Melalui simulasi interaktif, pengguna dapat mengeksplorasi berbagai skenario perubahan lingkungan, misalnya dampak peningkatan lahan terbangun atau pengurangan area hijau, secara visual dan intuitif. Hal ini sangat berguna bagi perencana kota dan pembuat kebijakan untuk menguji berbagai strategi mitigasi sebelum diterapkan di lapangan (Peng *et al.*, 2021). Selain itu, visualisasi adaptif mampu menyesuaikan tampilan dan informasi yang disajikan sesuai dengan kebutuhan pengguna, sehingga meningkatkan pemahaman dan keterlibatan *stakeholder* dalam perencanaan lingkungan.

Fenomena UHI telah menjadi perhatian utama di berbagai kota besar di dunia, seperti New York, Tokyo, dan Delhi, di mana peningkatan suhu akibat aktivitas *urbanisasi* menyebabkan dampak lingkungan dan kesehatan yang signifikan. Namun, keterlibatan pemangku kepentingan dalam proses mitigasi sering terhambat oleh keterbatasan dalam memahami kompleksitas spasial-temporal dari UHI, yang sebagian besar disebabkan oleh belum tersedianya alat visualisasi yang interaktif dan informatif (He *et al.*, 2022; Zhang *et al.*, 2023).



Kondisi serupa juga terjadi di sejumlah kota besar di Indonesia, seperti Jakarta, Surabaya, Bandung, dan Medan, yang menunjukkan tren peningkatan suhu permukaan akibat perubahan tutupan lahan, peningkatan kepadatan bangunan, serta rendahnya ruang terbuka hijau. Di kota-kota tersebut, tantangan utama dalam pengendalian UHI adalah kurangnya visualisasi yang dapat diterjemahkan secara langsung ke dalam kebijakan spasial, sehingga proses pengambilan keputusan oleh pemangku kepentingan menjadi kurang optimal (Nurwanda & Honjo, 2020; Rahman *et al.*, 2021).

Kondisi serupa mulai terlihat pula di Kota Malang, Jawa Timur, yang meskipun dikenal sebagai kota pendidikan dan budaya, kini tengah mengalami percepatan *urbanisasi* yang signifikan. Peningkatan pembangunan infrastruktur dan pertumbuhan penduduk telah memicu perubahan tata guna lahan yang pada akhirnya berkontribusi terhadap peningkatan intensitas UHI. Menanggapi tantangan ini, penelitian ini mengembangkan ruang virtual sebagai media visualisasi adaptif yang mengintegrasikan data penginderaan jauh dengan algoritma SVR untuk mensimulasikan kondisi UHI secara interaktif. Pendekatan ini diharapkan dapat menghasilkan model simulasi yang mampu memvisualisasikan dinamika distribusi suhu secara *real-time*, memberikan gambaran yang komprehensif mengenai dampak UHI, serta mendukung pengambilan keputusan dan strategi mitigasi yang lebih efektif (Kusumadewi *et al.*, 2023).

Pendekatan inovatif dalam visualisasi adaptif berbasis ruang virtual ini tidak hanya memberikan kontribusi signifikan pada pengembangan metode simulasi

UHI, tetapi juga membuka peluang untuk penerapan teknologi serupa pada permasalahan lingkungan perkotaan lainnya. Dengan menekankan interaktivitas dan akurasi prediksi, ruang virtual sebagai *platform* simulasi memungkinkan pengguna untuk mengevaluasi berbagai skenario kebijakan dengan lebih intuitif, sekaligus meningkatkan partisipasi *stakeholder* dalam perencanaan kota yang berkelanjutan.

Melalui penelitian ini, diharapkan dapat dikembangkan sebuah model prediktif yang mengintegrasikan data penginderaan jauh, analisis spasial, dan algoritma SVR. Model ini tidak hanya akan memberikan prediksi suhu permukaan yang akurat, tetapi juga menyajikan representasi visual yang interaktif, sehingga memudahkan pengambilan keputusan dalam perencanaan kota yang berkelanjutan. Pendekatan inovatif ini sejalan dengan tren global dalam pengelolaan kota pintar dan memiliki potensi besar untuk diterapkan di berbagai kota lain yang menghadapi permasalahan serupa (Furuya *et al.*, 2023).

Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan metode prediksi UHI yang lebih akurat, tetapi juga memberikan landasan bagi integrasi teknologi visualisasi dan kecerdasan buatan dalam pengelolaan lingkungan perkotaan. Hasil penelitian diharapkan dapat menjadi acuan bagi perencana kota, pembuat kebijakan, serta peneliti di bidang lingkungan dan *smart city*, guna menciptakan kota yang lebih nyaman, sehat, dan berkelanjutan.

## 1.2 Pernyataan Masalah

Berdasarkan pada latar belakang yang telah dijelaskan sebelumnya, rumusan masalah pada penelitian ini yaitu :

1. Bagaimana performa model SVR dalam memprediksi fenomena UHI di Kota Malang berdasarkan faktor-faktor penentunya?
2. Bagaimana tingkat *usability* dari visualisasi adaptif dalam lingkungan virtual yang dikembangkan untuk mensimulasikan hasil prediksi fenomena UHI di Kota Malang berdasarkan faktor-faktor penentunya?

## 1.3 Tujuan Penelitian

1. Menganalisis dan mengevaluasi performa model SVR dalam memprediksi fenomena UHI di Kota Malang berdasarkan faktor-faktor penentu yang diperoleh dari data penginderaan jauh satelit.
2. Mengevaluasi tingkat *usability* dari visualisasi adaptif dalam lingkungan virtual yang dikembangkan untuk mensimulasikan hasil prediksi fenomena UHI secara spasial di Kota Malang.

## 1.4 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah :

1. Data yang digunakan berdasarkan penginderaan jauh satelit melalui *Google Earth Engine* menggunakan Satelit Landsat-8 , Sentinel 2 dan SRTM.
2. Batas administrasi yang digunakan adalah 5 Kecamatan di Kota Malang yaitu Klojen, Blimbing, Lowokwaru, Sukun dan Kedungkandang.

3. Evaluasi performa metode SVR menggunakan *R-squared* ( $R^2$ ), *Mean Absolute Percentage Error* (MAPE). Pengukuran *usability* untuk visualisasi adaptif dalam lingkungan virtual menggunakan *SUS*.

### **1.5 Manfaat Penelitian**

Menyediakan platform simulasi interaktif yang dapat digunakan oleh perencana kota, akademisi, dan pembuat kebijakan dalam mengevaluasi dampak perubahan tata guna lahan terhadap distribusi suhu di Kota Malang.

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Fenomena UHI**

Fenomena UHI adalah suatu kondisi klimatologi dimana daerah pusat kota memiliki suhu yang lebih tinggi dari daerah pinggir kota (Nainggolan *et al.*, 2020). Faktor utama terbentuknya UHI adalah sebagian vegetasi digantikan oleh aspal dan beton untuk jalan, bangunan, dan infrastruktur lain yang diperlukan untuk mengakomodasi kebutuhan dan pertumbuhan populasi manusia. Permukaan yang tergantikan tersebut lebih banyak menyerap panas matahari dan juga lebih banyak memantulkannya, sehingga mengakibatkan suhu permukaan naik.

Wilayah *urban* akan cenderung mengalami peningkatan suhu dibandingkan dengan wilayah *sub-urban*. Khususnya, setelah matahari terbit suhu udara di daerah *rural* akan menyamai suhu udara di wilayah *urban*. Hal ini disebabkan wilayah *urban* memiliki tutupan bayangan oleh bangunan tinggi atau disebut dengan *urban canopy* dan melemahnya sinar matahari karena lapisan polusi yang terangkat yang mengakibatkan suhu udara meningkat lebih lambat pada pagi hari. Beberapa hasil kajian UHI teridentifikasi bahwa perbedaan suhu udara perkotaan lebih tinggi 0.02-1 °C dibandingkan dengan daerah di sekitarnya pada daerah tropis (Stewart *et al.*, 2021). Penelitian tentang UHI di beberapa kota besar di Indonesia dengan data satelit menunjukkan adanya perubahan temperatur yang merupakan salah satu indikasi adanya perubahan iklim, Hal ini ada hubungannya dengan perubahan lahan yang terjadi akibat *urbanisasi*. Di Bandung teramati

perluasan UHI daerah dengan suhu tinggi 30-35<sup>0</sup> C yang terletak pada kawasan terbangun di pusat kota per tahun kira-kira 12606 ha atau 4.47%, di Semarang 12174 ha atau 8.4%, di Surabaya 1512 ha atau 4.8%. Pertumbuhan kawasan terbangun di Bandung per tahun kurang lebih 1029 ha (0.36%), Semarang 1200 ha (0.83%), dan Surabaya 531.28 ha (1.69%) (Munawaroh, 2015).

## 2.2 Faktor-faktor yang mempengaruhi UHI

Di perkotaan, sebaran spasial LST tidak homogen namun menghadirkan heterogenitas spasial. Heterogenitas spasial LST di suatu kota biasanya bervariasi karena variasi sifat permukaan (F. Li *et al.*, 2023). Dalam literatur sebelumnya, telah diketahui bahwa komposisi lanskap, seperti permukaan kedap air, badan air, dan ruang hijau, sangat menentukan distribusi spasial LST di perkotaan (Liu *et al.*, 2021). Oleh karena itu, penting untuk memasukkan *Satellite-derived Environmental Variables* (SEV) sebagai faktor independen yang memberikan informasi mengenai berbagai aspek lingkungan di suatu wilayah dan dapat digunakan untuk analisis berbagai fenomena, termasuk fenomena UHI. Secara umum, SEV dapat dikategorikan menjadi 4 kategori seperti pada tabel 2.1 :

Tabel 2. 1 Faktor yang mempengaruhi UHI

Kategori	Contoh Variabel	Sumber data (satelit & band)	Peran terhadap UHI
Permukaan Kedap Air	NDBI, built-up, NDBaI %	Landsat-8 (Band SWIR, NIR, Red), Sentinel-2 (SWIR, NIR, Red)	Permukaan kedap air meningkatkan penyerapan panas
Vegetasi	NDVI, SAVI, tutupan vegetasi (%)	Landsat-8 (NIR, Red), Sentinel-2 (NIR, Red), MODIS (NIR, Red)	Vegetasi memberi efek pendinginan melalui evaporasi dan naungan
Air (Ruang Biru)	NDWI, MNDWI, suhu permukaan air	Landsat-8 (Green, NIR, SWIR), Sentinel-2, MODIS, reanalysis (ERA5 untuk	Badan air menurunkan temperatur lokal sehingga mampu mengurangi intensitas UHI di sekitarnya.

		parameter meteoro)	
Panas (Termal)	LST (dari band termal), suhu udara (obs)	Landsat-8 TIRS (band termal), MODIS (termal), pengukuran stasiun udara	Variabel utama; LST merepresentasikan kondisi termal permukaan yang menjadi indikator langsung UHI.

a. Permukaan Kedap Air

Permukaan Kedap Air merupakan kategori yang berkaitan dengan tutupan lahan, seperti jenis tutupan tanah, contohnya hutan, lahan pertanian, dan perkotaan dan karakteristik fisik permukaan tanah seperti tekstur tanah dan elevasi. Data ini diperoleh melalui citra satelit, dengan menggunakan satelit seperti *Landsat* dan *Sentinel-2*, yang memiliki berbagai band-band yang mendukung, seperti *Red*, *Green*, *Blue*, *NIR*, dan *SWIR*. Contoh variabel yang sering digunakan adalah NDBI dan NDBaI. Kedua variabel tersebut digunakan untuk menentukan dampak permukaan kedap air terhadap intensitas UHI. Berdasarkan penelitian sebelumnya oleh (Addas, 2023), didapatkan kesimpulan bahwa daerah yang memiliki proporsi permukaan kedap air yang tinggi maka nilai intensitas UHI pada daerah tersebut juga tinggi (Shi *et al.*, 2021).

b. Vegetasi

Vegetasi merupakan kategori yang berkaitan dengan keadaan vegetasi, seperti NDVI (*Normalized Difference Vegetation Index*), SAVI, luas tutupan vegetasi, dan parameter-parameter lainnya. Sejumlah satelit seperti MODIS, Landsat, dan Sentinel-2 digunakan untuk mengumpulkan data ini, dengan penggunaan *band* seperti *NIR* dan *Red*.

Parameter ini telah banyak digunakan pada penelitian sebelumnya untuk mengamati pengaruh tutupan vegetasi terhadap intensitas UHI. Telah terbukti

bahwa intensitas UHI yang rendah terjadi di tempat dengan tutupan vegetasi yang tinggi dan sebaliknya (J. Li *et al.*, 2021). Penelitian terbaru juga menemukan bahwa tutupan vegetasi mempunyai efek pendinginan yang signifikan di wilayah perkotaan (Sannigrahi *et al.*, 2018). Oleh karena itu, indeks vegetasi dapat menjadi parameter yang sangat membantu untuk memahami intensitas UHI di suatu wilayah tertentu (X. Zhang *et al.*, 2017).

c. Air

Kategori air melibatkan pemantauan variabel lingkungan yang berkaitan dengan kualitas dan kuantitas air di suatu daerah. Contoh variabel yang sering digunakan adalah NDWI, MNDWI, suhu permukaan air, konsentrasi sedimen, dan tingkat keasaman. Satelit yang sering digunakan untuk parameter ini adalah Landsat, Modis, ERA5, dan Sentinel-2. Pada penelitian sebelumnya menunjukkan bahwa badan air atau ruang biru secara signifikan mengurangi suhu. Dengan demikian, Perairan atau ruang biru juga memiliki dampak yang signifikan terhadap intensitas UHI (Odindi *et al.*, 2015).

d. Panas

Kategori panas melibatkan pemantauan variabel lingkungan yang berhubungan dengan suhu atmosfer, seperti suhu permukaan tanah dan suhu udara. Dalam hal ini, satelit yang sering digunakan adalah satelit seperti Landsat dan MODIS. Contohnya pada satelit Landsat, memiliki band termal yang digunakan dapat digunakan untuk mengukur LST.

LST adalah isu yang memberikan kontribusi terkuat yang mempengaruhi intensitas UHI (Bokaie *et al.*, 2016). Hal ini sebagian besar dipengaruhi oleh



konfigurasi dan komposisi permukaan tanah di suatu wilayah. Misalnya, sifat kedap air pada permukaan berhubungan positif dengan LST, sedangkan vegetasi dan badan air berhubungan negatif dengan LST (X. Zhang *et al.*, 2017). Dengan demikian, perubahan komposisi biofisik sangat mempengaruhi LST (X. Zhang *et al.*, 2017). Kantong dengan LST tinggi di perkotaan ditandai dengan intensitas UHI yang tinggi (Mathew *et al.*, 2016). Dengan demikian, LST telah dianggap sebagai faktor dominan yang mempengaruhi intensitas UHI. Selain suhu permukaan tanah, suhu udara juga sering digunakan untuk memberikan informasi tentang perubahan iklim, pola suhu, dan distribusi energi termal di suatu wilayah.

### 2.3 Machine Learning

Perkembangan *Machine Learning* (ML) telah memberikan kontribusi yang signifikan dalam bidang pemodelan lingkungan, termasuk dalam analisis dan prediksi fenomena UHI. Pendekatan ML mampu mengenali pola kompleks serta hubungan *nonlinier* antara berbagai variabel lingkungan seperti suhu permukaan, tutupan lahan, kelembapan, kerapatan vegetasi, ketinggian, serta parameter spasial lainnya yang sulit ditangkap menggunakan metode statistik konvensional. Selain itu, ML memungkinkan integrasi data multisumber seperti citra satelit misalnya Landsat, Sentinel, atau MODIS, data geospasial, serta data hasil observasi lapangan untuk membangun model prediktif yang lebih akurat dan adaptif terhadap kondisi dinamis perkotaan. Dalam penelitian-penelitian sebelumnya, berbagai algoritma ML telah digunakan untuk memodelkan dan memprediksi intensitas UHI, antara lain regresi linier berganda, model berbasis pohon

keputusan seperti *Random Forest* (RF), model ensemble learning seperti *Gradient Boosting Machine* (GBM), serta model berbasis Jaringan Saraf Tiruan.

Model regresi linier masih sering digunakan karena kemudahan interpretasinya, namun cenderung terbatas dalam menangani hubungan *nonlinier* antarvariabel lingkungan. Sementara itu, pendekatan berbasis pohon keputusan dan ensemble learning seperti RF terbukti efektif dalam menangani data yang heterogen dan kompleks, tetapi hasilnya sering kali bersifat “kotak hitam” karena sulit diinterpretasikan secara matematis. Di sisi lain, model berbasis jaringan saraf atau deep learning memiliki kemampuan tinggi dalam mendeteksi pola *nonlinier* yang rumit, namun memerlukan dataset besar, sumber daya komputasi yang tinggi, serta waktu pelatihan yang lama, yang tidak selalu efisien untuk data berukuran menengah atau dalam konteks integrasi cepat dengan sistem visualisasi spasial interaktif. Dalam konteks tersebut, metode *Support Vector Machine* (SVM) dan varian regresinya, yaitu SVR, muncul sebagai pendekatan yang seimbang antara akurasi, efisiensi, dan interpretabilitas. SVR mampu menangani hubungan *nonlinier* antarvariabel menggunakan fungsi kernel tanpa memerlukan arsitektur jaringan kompleks seperti pada deep learning. Pendekatan ini memanfaatkan prinsip margin maximization dan konsep kernel trick untuk memproyeksikan data ke ruang berdimensi lebih tinggi sehingga pola hubungan antar fitur dapat dimodelkan secara linier di ruang tersebut. Keunggulan lain SVR adalah kemampuannya untuk menghasilkan prediksi yang stabil pada dataset berukuran sedang dan berdimensi tinggi, karakteristik umum pada data UHI yang

terdiri atas kombinasi indeks lingkungan seperti NDVI, NDBI, NDWI, NDMI, UDI serta data topografi atau geospasial lainnya.

Dalam penelitian ini, pendekatan machine learning tidak hanya digunakan untuk melakukan estimasi kuantitatif terhadap suhu permukaan atau intensitas UHI, tetapi juga sebagai dasar pengembangan visualisasi interaktif berbasis *game engine*. Hasil prediksi yang dihasilkan oleh model SVR diintegrasikan ke dalam platform Unity3D untuk membangun representasi spasial dan temporal yang realistis mengenai sebaran panas di kawasan perkotaan. Integrasi ini memungkinkan pengguna atau pemangku kebijakan untuk mengeksplorasi dinamika perubahan suhu permukaan secara imersif dan intuitif melalui lingkungan tiga dimensi. Pemilihan SVR dalam penelitian ini didasarkan pada beberapa pertimbangan utama, antara lain kinerja model yang stabil pada data penginderaan jauh, kemampuan generalisasi yang baik dalam memprediksi nilai kontinu seperti suhu permukaan, efisiensi komputasi yang mendukung integrasi langsung dengan sistem visualisasi tanpa memerlukan perangkat keras komputasi berat, serta kemampuannya menangani hubungan *nonlinier* antar fitur menggunakan fungsi kernel. Dengan demikian, penggunaan SVR bukan hanya didasari oleh keunggulan matematisnya dalam pemodelan regresi *nonlinier*, tetapi juga oleh kesesuaiannya dalam menghasilkan prediksi yang presisi, stabil, dan dapat divisualisasikan secara spasial-interaktif dalam lingkungan berbasis game visualization, sehingga mendukung proses analisis dan pengambilan keputusan berbasis data dalam konteks mitigasi dan pemantauan fenomena UHI.

## 2.4 SVR

Support Vector Machine adalah salah satu model pembelajaran untuk klasifikasi dan regresi. Gabungan Vector Machine untuk regresi secara khusus dikatakan sebagai SVR. SVR dapat berupa linier atau non-linier menggunakan fungsi kernel masing-masing (Kavitha *et al.*, 2017). Namun, tidak seperti klasifikasi SVM yang menghasilkan output biner (yaitu label kelas), SVR menangani masalah regresi yang memungkinkan estimasi fungsi bernilai nyata (misalnya, skor kontinu dalam skala klinis). SVR menerapkan ide dasar dari SVM, yaitu mesin kernel jarang yang melakukan klasifikasi menggunakan hyperplane yang didefinisikan oleh beberapa vektor pendukung. Hasilnya, optimasi dalam SVR direpresentasikan dalam bentuk support vector (sekumpulan kecil sampel data training), dimana solusi optimasi tidak bergantung pada dimensi data input, tetapi hanya bergantung pada jumlah support vector. SVR memiliki kelebihan tambahan jika dibandingkan dengan metode regresi lainnya. Dengan penggunaan kernel, SVR dapat memberikan cara yang efisien untuk menangani masalah regresi *nonlinier* dengan memproyeksikan fitur asli ke dalam ruang kernel di mana data dapat didiskriminasikan secara linier (Orrù *et al.*, 2012).

SVR dikembangkan oleh Vapnik dan rekan-rekannya dengan memperluas algoritma SVM. Dalam pembelajaran mesin, SVM terkenal dengan kinerjanya yang baik untuk menangani data berdimensi tinggi. SVM didasarkan pada kerangka teori pembelajaran statistik (teori Vapnik-Chervonenkis [VC]) dan menawarkan pendekatan yang berprinsip pada masalah pembelajaran mesin karena landasan matematika ini. Ide dasar SVM awalnya diusulkan pada tahun

60-an oleh Vapnik dkk. (1963, 1964), yang kemudian algoritmanya banyak dikembangkan pada dekade-dekade berikutnya. Keseluruhan sistem SVM secara resmi dipublikasikan pada tahun 1992 untuk klasifikasi dan kemudian untuk regresi.

Dalam klasifikasi SVM, setiap sampel yang diberi label diperlakukan sebagai titik data dalam ruang fitur multidimensi, dan hyperplane dalam ruang fitur ini dihitung untuk mengklasifikasikan sampel pelatihan sebanyak mungkin dengan benar. Sampel baru kemudian diklasifikasikan berdasarkan sisi hyperplane mana yang berada di ruang fitur multidimensi. Untuk menemukan hyperplane yang baik, pengoptimalan dilakukan dengan memaksimalkan margin antara vektor pendukung yaitu, titik data yang paling dekat dengan hyperplane. Untuk regresi data, alih-alih menemukan hyperplane yang dapat memisahkan sampel pelatihan, SVR memperkenalkan fungsi loss  $\epsilon$ -insensitive untuk menghitung hyperplane sedemikian rupa sehingga nilai respons yang diprediksi dari sampel pelatihan memiliki deviasi paling banyak  $\epsilon$  dari nilai respons yang diamati. Hyperplane ditambah  $\epsilon$  mendefinisikan sebuah tabung atau pita  $\epsilon$ -insensitif untuk menghitung batas generalisasi untuk regresi. Optimasi dilakukan dengan meminimalkan tabung  $\epsilon$ -insensitif menjadi serata mungkin sambil berisi sebagian besar sampel pelatihan. Dalam hal ini, hyperplane direpresentasikan dalam bentuk beberapa vektor pendukung, yaitu sampel pelatihan yang berada di luar batas  $\epsilon$ -insensitive tube. Sebagai hasil dari pelatihan SVR, sebuah model regresi dipelajari untuk memprediksi keluaran respons untuk sampel baru. Dalam penelitian ini SVR digunakan untuk memprediksi nilai variabel-variabel yang digunakan untuk

memberikan nilai hasil prediksi yang baik berupa nilai aktual dan hasil prediksi yang dapat diminimalkan tingkat kesalahannya dalam prediksi.

Salah satu penelitian yang memanfaatkan metode ini penelitian Dhaloiya et al. (2023), SVR dibandingkan dengan metode regresi linier dalam memprediksi nilai evapotranspirasi di India. Penelitian ini menggunakan varian *Least Squares SVR* (LS-SVM) yang bertujuan untuk mengurangi kompleksitas komputasi dalam model tradisional SVR. Input yang digunakan mencakup indeks vegetasi dan kelembaban seperti NDVI, NDWI, dan NDMI. Hasil pengujian menunjukkan bahwa SVR secara signifikan menghasilkan prediksi yang lebih akurat daripada metode linier, memperkuat bukti bahwa SVR sangat efektif dalam menangani data penginderaan jauh yang memiliki hubungan non-linear antar variabel input (Dhaloiya et al., 2023).

Sementara itu, Mohite et al. (2022) menerapkan SVR dalam konteks yang berbeda, yaitu untuk melakukan downscaling data Vegetation Optical Depth (VOD) dari resolusi kasar 25 km ke resolusi lebih tinggi 1 km. Studi ini bertujuan untuk mengintegrasikan data dengan simulasi resolusi tinggi seperti visualisasi suhu kota atau analisis kelembaban vegetatif spasial. SVR terbukti efektif dalam proses ini, terutama untuk data bulan Januari yang menunjukkan performa terbaik. Hasil ini menunjukkan bahwa SVR tidak hanya dapat digunakan dalam prediksi langsung, tetapi juga sebagai alat peningkatan resolusi data penginderaan jauh untuk aplikasi lanjutan (Mohite et al., 2022).

Berbagai penelitian di Indonesia juga telah menerapkan metode SVR untuk memprediksi fenomena UHI. Kusumadewi et al. (2025) mengembangkan pendekatan SVR multivariabel yang mengombinasikan indikator lingkungan, sosial, dan kepadatan *urban* untuk memprediksi dinamika UHI di Kota Malang. Penelitian ini menghasilkan nilai  $R^2 = 0,78$  dan  $MAPE = 0,46 \%$ , yang menegaskan bahwa penambahan faktor sosial-spasial dapat meningkatkan presisi model dalam memetakan pola termal di wilayah perkotaan.

## 2.5 Visualisasi Berbasis Machine Learning

Visualisasi data dalam konteks ilmiah dan lingkungan tidak lagi terbatas pada penyajian statis dalam bentuk grafik dua dimensi, tetapi telah berkembang menjadi bentuk visualisasi yang dinamis, adaptif, dan interaktif. Visualisasi adaptif mengacu pada sistem visual yang dapat menyesuaikan representasinya berdasarkan kondisi data, konteks lingkungan, atau interaksi pengguna. Dalam pendekatan ini, elemen visual seperti warna, tekstur, ukuran, dan intensitas cahaya dapat berubah secara otomatis mengikuti dinamika data atau masukan pengguna, sehingga meningkatkan pemahaman terhadap fenomena yang ditampilkan. Konsep ini sejalan dengan pandangan Bimber dan Raskar (2004), yang menyatakan bahwa visualisasi modern tidak lagi bersifat pasif, melainkan harus bersifat responsive dan kontekstual, sehingga pengalaman pengguna menjadi lebih imersif dan intuitif.

Perkembangan teknologi *interactive visualization* juga mendorong munculnya paradigma baru dalam komunikasi data ilmiah, di mana pengguna tidak hanya menjadi pengamat, tetapi juga partisipan aktif dalam proses eksplorasi

data. Visualisasi interaktif memungkinkan manipulasi langsung terhadap parameter data, seperti mengubah skenario lingkungan, menyesuaikan skala waktu, atau memfilter area tertentu untuk melihat variasi hasil secara *real-time*. Dalam konteks perencanaan kota dan analisis lingkungan, interaktivitas ini memiliki peran penting untuk mendukung proses pengambilan keputusan berbasis bukti (*evidence-based decision making*), karena pengguna dapat memahami keterkaitan spasial dan temporal dari fenomena kompleks seperti UHI.

Integrasi machine learning dengan visualisasi adaptif menghasilkan paradigma baru yang disebut visualisasi prediktif adaptif. Dalam pendekatan ini, model ML seperti SVR digunakan untuk menghasilkan data prediktif yang kemudian divisualisasikan secara dinamis. Hasil prediksi suhu permukaan, indeks termal, atau intensitas panas kota tidak hanya ditampilkan sebagai peta statis, tetapi berubah secara adaptif berdasarkan input pengguna, waktu simulasi, atau skenario lingkungan yang dipilih. Dengan demikian, visualisasi adaptif berbasis ML bukan hanya menampilkan hasil analisis, melainkan juga memfasilitasi eksplorasi “*what-if scenarios*” dalam lingkungan yang realistis dan mudah dipahami.

Pendekatan ini sangat relevan dalam konteks UHI karena fenomena UHI bersifat spasial, dinamis, dan dipengaruhi oleh banyak variabel *nonlinier* seperti kepadatan bangunan, vegetasi, tutupan lahan, serta kondisi atmosfer. Melalui model SVR yang telah dilatih menggunakan data penginderaan jauh misalnya NDVI, NDBI, NDWI, dan LST, sistem dapat memprediksi perubahan suhu permukaan berdasarkan variabel spasial dan temporal tertentu. Selanjutnya, hasil



prediksi tersebut dapat diterjemahkan ke dalam bentuk visual adaptif, misalnya dengan perubahan warna permukaan kota berdasarkan tingkat panas, atau perubahan intensitas cahaya yang merepresentasikan perbedaan suhu antar zona. Visualisasi semacam ini memungkinkan pengguna untuk secara intuitif memahami distribusi panas perkotaan dan mengevaluasi dampak perubahan lingkungan terhadap kenyamanan termal kota.

## 2.6 Visualisasi UHI

Visualisasi UHI berbasis teknologi digital telah berkembang menjadi alat penting dalam komunikasi ilmiah, edukasi, dan pengambilan keputusan. Berbagai studi telah menunjukkan bahwa pendekatan visual yang imersif mampu meningkatkan pemahaman spasial serta keterlibatan emosional pengguna terhadap isu iklim dan *urbanisasi*. Penelitian Dhunnoo et al. (2023) menunjukkan bahwa penggunaan *immersive* VR dalam edukasi iklim mampu memberikan pengalaman langsung terhadap simulasi suhu ekstrem di lingkungan virtual. Hasil studi tersebut menegaskan bahwa VR tidak hanya meningkatkan pemahaman konseptual pengguna terhadap perubahan iklim, tetapi juga memicu reaksi emosional yang memperkuat kesadaran dan empati terhadap dampaknya. Temuan ini menegaskan potensi besar VR sebagai media komunikasi ilmiah yang efektif dan berdampak emosional dalam kampanye mitigasi iklim.

Selain VR, pendekatan berbasis peta dua dimensi juga masih relevan dalam memvisualisasikan fenomena UHI. Techer et al. (2023) mengembangkan sistem visualisasi prediktif berbasis 2D Maps untuk menggambarkan kerentanan termal kota dalam skenario masa depan. Melalui peta suhu dinamis yang dihasilkan dari

model prediksi spasial, pengguna dapat menelusuri dampak kenaikan suhu pada area tertentu serta mengevaluasi strategi adaptasi yang sesuai. Pendekatan ini dinilai efektif untuk mendukung pengambilan keputusan dalam perencanaan kota adaptif. Penelitian lain oleh Farhan et al. (2024) memperluas konsep ini dengan mengintegrasikan Google Earth Engine (GEE) dan teknologi interactive mapping untuk membangun sistem visualisasi prediktif spasial yang lebih interaktif. Dalam sistem ini, pengguna dapat menjelajahi berbagai skenario suhu masa depan berdasarkan data Land Use Land Cover (LULC) dan Land Surface Temperature (LST) yang diproses otomatis menggunakan model ML. Keunggulan utama pendekatan ini adalah kemampuannya menggabungkan machine learning, remote sensing, dan teknologi pemetaan dalam satu ekosistem terpadu untuk mendukung mitigasi risiko iklim secara komprehensif.

## **2.7 Visualisasi Adaptif pada Ruang Virtual**

Visualisasi adaptif pada ruang virtual merepresentasikan evolusi lanjutan dari visualisasi interaktif, di mana sistem mampu menyesuaikan tampilan dan perilaku elemen visual berdasarkan konteks, skenario, atau interaksi pengguna. Elemen-elemen visual seperti warna, pencahayaan, bentuk, atau ukuran dapat berubah secara dinamis sesuai kondisi simulasi, sehingga menghasilkan pengalaman yang lebih realistis dan kontekstual. Menurut Bimber dan Raskar (2024), teknologi VR memungkinkan penciptaan lingkungan yang dapat menyesuaikan diri dengan kebutuhan dan kebiasaan pengguna, memberikan pengalaman visual yang personal dan intuitif. Penerapan konsep ini telah banyak dieksplorasi dalam berbagai bidang, termasuk perencanaan kota dan mitigasi

perubahan iklim. Sebagai contoh, teknologi Augmented Reality (AR) digunakan dalam proyek infrastruktur di Tampa Bay, Florida, untuk menampilkan model 3D bangunan tahan banjir yang di-*overlay* langsung ke dunia nyata melalui perangkat AR. Studi yang dilaporkan oleh Business Insider (2025) menunjukkan bahwa pendekatan ini membantu masyarakat memahami secara visual dampak proyek terhadap lingkungan dan meningkatkan partisipasi publik dalam proses perencanaan. Penelitian oleh Kato et al. (2025) serta Pham et al. (2025) juga menunjukkan bahwa visualisasi adaptif dalam ruang virtual dapat memperkuat perencanaan kota yang lebih responsif terhadap perubahan sosial, ekonomi, dan lingkungan.

Dalam konteks penelitian ini, visualisasi adaptif pada ruang virtual diimplementasikan dengan mengintegrasikan hasil prediksi SVR ke dalam Unity3D-based environment. Hasil prediksi suhu permukaan yang dihasilkan oleh model SVR digunakan untuk memodifikasi elemen visual secara *real-time* misalnya, perubahan warna bangunan, pencahayaan lingkungan, atau intensitas cahaya matahari dalam dunia virtual yang mencerminkan tingkat panas aktual di area tersebut. Dengan demikian, visualisasi ini tidak hanya menyajikan data statis, tetapi juga menghadirkan representasi adaptif yang dapat berubah mengikuti dinamika hasil prediksi. Hal ini memungkinkan pengguna untuk menjelajahi berbagai skenario spasial-temporal secara imersif, memahami pola distribusi panas perkotaan secara mendalam, dan mengevaluasi strategi mitigasi UHI dengan pendekatan berbasis pengalaman.

## 2.8 Integrasi API untuk Penghubung Data Nyata, Model Prediktif, dan Visualisasi Adaptif

Application Programming Interface (API) berperan sebagai jembatan penghubung antara berbagai komponen sistem yang berbeda, baik dari sisi sumber data, model komputasi, maupun media visualisasi. Dalam konteks penelitian berbasis *urban computing* dan analisis UHI, API menjadi tulang punggung yang memungkinkan aliran data *real-time* antara lapisan data (data lingkungan atau satelit), lapisan model prediktif (machine learning atau SVR), dan lapisan visualisasi interaktif (misalnya platform berbasis Unity atau WebGL). Menurut Fielding (2000), konsep dasar dari API modern khususnya yang berbasis arsitektur REST (Representational State Transfer) adalah menyediakan antarmuka komunikasi yang efisien dan terstandarisasi, di mana berbagai sistem dapat saling bertukar data tanpa harus memahami implementasi internal masing-masing.

Dalam sistem prediksi dan visualisasi iklim, seperti UHI, API memungkinkan data yang diperoleh dari platform penginderaan jauh seperti Google Earth Engine (GEE) untuk dikirim secara langsung ke backend server, diolah menggunakan model pembelajaran mesin seperti SVR, kemudian hasilnya dikirim kembali ke sistem visualisasi untuk divisualisasikan secara dinamis. Pendekatan ini menciptakan alur data yang bersifat dua arah dan adaptif di mana perubahan input dari lingkungan nyata dapat segera tercermin pada tampilan virtual. Studi oleh Gubbi et al. (2013) dan Lee et al. (2021) menekankan bahwa API memainkan peran sentral dalam ekosistem Internet of Things (IoT) dan smart

environment, karena memungkinkan integrasi antara pengumpulan data, analisis cerdas, dan presentasi hasil yang dapat diakses pengguna akhir.

Dari perspektif sistem, arsitektur API yang baik tidak hanya berfungsi sebagai saluran data, tetapi juga sebagai orchestrator yang mengelola seluruh proses prediksi dan visualisasi. Setiap permintaan dari klien (misalnya Unity) diproses melalui tahapan validasi input, pengambilan data dari sumber eksternal, pembentukan feature collection, prediksi model, dan penyusunan output dalam format yang dapat langsung divisualisasikan. Menurut Richardson & Ruby (2008), desain API yang modular dan asinkron menjadi kunci dalam memastikan sistem dapat beroperasi secara responsif, terutama ketika menangani permintaan dari banyak pengguna atau sumber data berskala besar.

Dalam konteks visualisasi adaptif berbasis VR, integrasi API menjadi semakin penting karena memungkinkan visualisasi virtual untuk “bernapas” dan beradaptasi terhadap perubahan nyata. Misalnya, nilai suhu atau indeks vegetasi yang diperbarui dari GEE dapat secara otomatis memperbarui warna atau material pada model 3D di Unity, menghasilkan efek *heatmap* dinamis yang menggambarkan fluktuasi spasial suhu permukaan. Prinsip ini sejalan dengan konsep *real-time* adaptive visualization yang dikemukakan oleh Munzner (2014), di mana visualisasi tidak lagi bersifat statis, melainkan mampu bereaksi terhadap perubahan konteks data dan perilaku pengguna.

Lebih jauh lagi, API juga memungkinkan terjadinya integrasi lintas platform antara data ilmiah, sistem prediksi berbasis machine learning, dan

teknologi imersif seperti VR atau Augmented Reality (AR). Studi oleh Pham et al. (2025) menunjukkan bahwa penerapan API dalam lingkungan virtual dapat meningkatkan tingkat interaktivitas, personalisasi, dan keterlibatan pengguna, karena setiap perubahan data eksternal dapat langsung diproyeksikan dalam dunia virtual secara visual. Dengan demikian, API bukan hanya berfungsi sebagai data connector, tetapi juga sebagai penggerak utama (enabler) dari sistem visualisasi cerdas dan responsif yang menggabungkan realitas digital dan fisik.

Dari sisi teknis, penggunaan *framework* modern seperti FastAPI memberikan keuntungan dalam hal performa, kemudahan integrasi, dan dukungan terhadap komunikasi asinkron. Menurut Tiangolo (2022), FastAPI dirancang dengan dukungan penuh terhadap asynchronous I/O operations yang sangat sesuai untuk pemrosesan data *real-time* dan penghubungan ke model pembelajaran mesin yang telah diserialisasi dalam format .pkl. Dalam konteks penelitian UHI, pendekatan ini memungkinkan proses prediksi suhu permukaan tanah dilakukan secara otomatis dan cepat setiap kali ada permintaan baru dari Unity, tanpa menghambat kinerja visualisasi 3D.

Integrasi API juga membuka peluang bagi sistem untuk terus berkembang ke arah data-driven city simulation, di mana kondisi lingkungan perkotaan dapat dipantau dan divisualisasikan secara langsung melalui dunia virtual yang adaptif. Misalnya, dengan menghubungkan API ke layanan cuaca *real-time* seperti Open-Meteo atau Copernicus Climate Data Store, sistem dapat menampilkan prediksi suhu terkini dalam model 3D Kota Malang dan menyesuaikan tampilan sesuai perubahan cuaca aktual. Dengan demikian, kombinasi antara API, machine

learning, dan VR membentuk ekosistem cerdas yang tidak hanya menyajikan data, tetapi juga memungkinkan eksplorasi dan pemahaman yang lebih mendalam terhadap fenomena spasial dan termal di lingkungan perkotaan.

## 2.9 Evaluasi Performa

Evaluasi performa merupakan tahap penting dalam setiap penelitian yang melibatkan pengembangan model prediktif maupun sistem visualisasi interaktif. Evaluasi dilakukan untuk memastikan bahwa model maupun sistem yang dibangun tidak hanya berfungsi secara teknis, tetapi juga memiliki akurasi, reliabilitas, dan tingkat penerimaan pengguna yang tinggi. Dalam penelitian ini, evaluasi dilakukan melalui dua pendekatan utama: (1) evaluasi performa model prediksi suhu permukaan tanah berbasis SVR menggunakan indikator Mean Absolute Percentage *Error* (MAPE) dan Coefficient of Determination ( $R^2$ ), serta (2) evaluasi performa sistem visualisasi adaptif berbasis VR menggunakan SUS untuk menilai aspek kegunaan dan pengalaman pengguna.

### a. Evaluasi Performa Model Prediktif

Dalam ranah machine learning, pemilihan metrik evaluasi menjadi aspek krusial karena menentukan bagaimana kualitas hasil prediksi diinterpretasikan. Salah satu metrik yang paling banyak digunakan dalam penelitian berbasis regresi adalah Mean Absolute Percentage *Error* (MAPE). MAPE mengukur rata-rata persentase kesalahan absolut antara nilai prediksi dan nilai aktual, yang dinyatakan dalam satuan persentase. Menurut Hyndman dan Koehler (2006), keunggulan utama MAPE terletak pada interpretasinya yang mudah dipahami

karena hasilnya secara langsung menunjukkan seberapa besar kesalahan rata-rata dalam bentuk persentase. Nilai MAPE yang kecil menandakan bahwa model memiliki tingkat kesalahan prediksi yang rendah dan, dengan demikian, reliabilitas prediksinya tinggi.

Selain MAPE, penelitian ini juga menggunakan Coefficient of Determination ( $R^2$ ) untuk mengukur proporsi variasi data aktual yang dapat dijelaskan oleh model prediktif.  $R^2$  memberikan indikasi seberapa baik model mengikuti pola distribusi data sebenarnya. Menurut Draper dan Smith (1998), nilai  $R^2$  yang mendekati 1 menunjukkan bahwa model memiliki kemampuan penjelasan yang tinggi terhadap data, sedangkan nilai mendekati 0 menunjukkan bahwa model gagal menjelaskan variasi data yang diamati. Penggunaan kombinasi MAPE dan  $R^2$  memberikan pandangan yang lebih komprehensif terhadap performa model. MAPE berfokus pada kesalahan relatif, sedangkan  $R^2$  mengevaluasi kekuatan hubungan antara hasil prediksi dan nilai sebenarnya.

Beberapa metrik alternatif yang juga sering digunakan untuk mengevaluasi model regresi meliputi Root Mean Square *Error* (RMSE), Mean Absolute *Error* (MAE), dan Normalized Root Mean Square *Error* (NRMSE). RMSE sering dipilih karena sensitif terhadap kesalahan besar dan dapat menunjukkan deviasi rata-rata prediksi terhadap nilai aktual (Chai & Draxler, 2014). MAE memberikan ukuran kesalahan absolut rata-rata tanpa memperhatikan arah kesalahan, sedangkan NRMSE menormalkan nilai RMSE terhadap rentang data, sehingga hasilnya lebih mudah dibandingkan antar model. Namun, dalam konteks penelitian ini, MAPE dan  $R^2$  dinilai lebih sesuai karena dua alasan utama yaitu



nilai MAPE memberikan gambaran yang mudah diinterpretasikan untuk komunikasi hasil kepada pemangku kepentingan non-teknis seperti perencana kota dan pembuat kebijakan, dan nilai  $R^2$  menunjukkan keterkaitan linear dan non-linear antara data hasil penginderaan jauh dan hasil estimasi model SVR yang berbasis kernel RBF. Kombinasi kedua metrik ini mencerminkan keseimbangan antara akurasi numerik dan kemampuan model menjelaskan pola spasial suhu permukaan di kawasan perkotaan.

### **b. Evaluasi Performa Sistem Visualisasi Adaptif**

Selain akurasi model, keberhasilan sistem juga diukur dari seberapa baik pengguna dapat berinteraksi dan memahami informasi yang disajikan melalui antarmuka visual. Dalam konteks visualisasi adaptif berbasis VR, aspek *usability* menjadi faktor utama untuk menilai pengalaman pengguna. Salah satu instrumen yang paling banyak digunakan untuk mengevaluasi kegunaan sistem interaktif adalah SUS yang dikembangkan oleh Brooke (1996). SUS terdiri dari 10 butir pernyataan dengan skala Likert 5 poin, dan menghasilkan skor antara 0 hingga 100 yang merepresentasikan persepsi pengguna terhadap kemudahan penggunaan, efisiensi, dan kepuasan terhadap sistem.

SUS memiliki keunggulan dalam hal kesederhanaan, validitas lintas domain, dan kemampuannya digunakan pada berbagai jenis sistem, termasuk aplikasi web, sistem perangkat lunak, maupun lingkungan virtual (Bangor *et al.*, 2008). Dalam konteks penelitian ini, SUS digunakan untuk mengevaluasi tingkat kenyamanan dan kemudahan pengguna dalam memahami visualisasi 3D adaptif

yang menampilkan hasil prediksi UHI. Skor SUS yang tinggi menunjukkan bahwa sistem VR mudah digunakan, intuitif, dan memberikan pengalaman interaktif yang positif. Pengalaman interaktif yang positif tidak hanya diukur dari tingkat kemudahan penggunaan, tetapi juga dari beban kognitif, kualitas pengalaman emosional, dan tingkat kepuasan pengguna secara keseluruhan. Selain SUS, terdapat pula beberapa metode evaluasi *usability* lainnya seperti pada tabel 2.2 yaitu NASA Task Load Index (NASA-TLX), User Experience Questionnaire (UEQ), dan Post-Study System *Usability* Questionnaire (PSSUQ).

Tabel 2. 2 Evaluasi Performa Sistem Visualisasi Adaptif

<b>Instrumen</b>	<b>Fokus</b>	<b>Kelebihan</b>	<b>Penggunaan</b>
SUS	<i>Usability</i> umum	Cepat, lintas-domain, mudah dibandingkan	Evaluasi awal & perbandingan <i>usability</i>
NASA-TLX	Beban kerja kognitif	Mendalam untuk <i>workload</i>	Jika tugas kognitif kompleks diuji
UEQ	Pengalaman pengguna	Mengukur aspek emosional & estetika	Untuk evaluasi pengalaman keseluruhan
PSSUQ	Kepuasan pasca penggunaan	Menilai performa sistem & kepuasan	Setelah sesi penggunaan panjang

NASA-TLX, dikembangkan oleh Hart dan Staveland (1988), digunakan untuk menilai beban kerja mental pengguna selama berinteraksi dengan sistem; metode ini sangat berguna ketika sistem melibatkan banyak aktivitas kognitif kompleks. Dalam lima tahun terakhir, NASA-TLX banyak digunakan pada penelitian sistem VR dan simulasi. Sebagai contoh, Zhang et al. (2021) menggunakan NASA-TLX untuk mengevaluasi beban kerja pengguna dalam simulasi VR navigasi kota, dan hasilnya menunjukkan bahwa *workload* yang tinggi dapat menurunkan kinerja interaksi. Selain itu, Kim & Lee (2023)

menggunakan NASA-TLX pada sistem berbasis AI untuk menganalisis beban mental dalam proses pengambilan keputusan otomatis.

UEQ, yang dikembangkan oleh Laugwitz et al. (2008), mengevaluasi aspek pengalaman pengguna secara lebih luas, termasuk daya tarik visual, efisiensi, dan stimulasi emosional. Banyak penelitian game edukasi dan aplikasi interaktif terbaru mengadopsi UEQ. Misalnya, Santos et al. (2020) menerapkan UEQ pada game edukasi berbasis mobile dan menemukan bahwa aspek daya tarik (*attractiveness*) dan stimulasi memiliki pengaruh besar terhadap motivasi belajar. Selain itu, Müller & Schäfer (2022) menggunakan UEQ untuk menilai sistem *immersive environment* dan menunjukkan bahwa aspek hedonis sangat menentukan keberlanjutan penggunaan.

Sementara itu, PSSUQ (Lewis, 1995) lebih berfokus pada kepuasan pengguna terhadap performa sistem setelah sesi penggunaan selesai. Metode ini juga tetap relevan dalam penelitian terbaru. Contohnya, Rahman et al. (2019) menggunakan PSSUQ untuk mengevaluasi prototipe aplikasi web pelayanan publik dan menunjukkan bahwa kualitas informasi dan kemudahan navigasi menjadi faktor utama kepuasan. Selain itu, Li & Chen (2022) mengombinasikan PSSUQ dengan SUS dalam evaluasi sistem e-learning untuk memperoleh gambaran menyeluruh terhadap *usability* dan kepuasan pengguna.

Namun, dalam konteks penelitian ini, pemilihan SUS dianggap paling tepat karena beberapa pertimbangan metodologis. Pertama, SUS memiliki struktur sederhana yang memungkinkan pengumpulan umpan balik secara cepat tanpa

mengganggu alur interaksi pengguna di dalam lingkungan VR. Kedua, SUS terbukti efektif untuk sistem dengan basis antarmuka adaptif dan interaktif yang masih dalam tahap pengembangan (Sauro & Lewis, 2012). Ketiga, instrumen SUS telah banyak digunakan dalam evaluasi sistem berbasis visualisasi ilmiah dan VR (Albert & Tullis, 2013), sehingga hasilnya dapat dibandingkan dengan studi sejenis secara lebih obyektif.

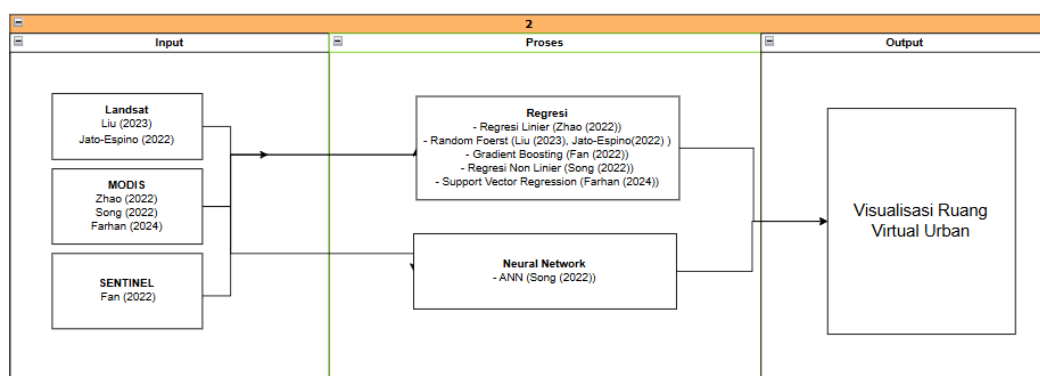
Dengan demikian, kombinasi antara evaluasi kuantitatif berbasis metrik model (MAPE dan  $R^2$ ) dan evaluasi kualitatif berbasis persepsi pengguna (SUS) memberikan kerangka penilaian yang menyeluruh. Pendekatan ini tidak hanya memastikan keakuratan prediksi model dalam merepresentasikan fenomena UHI secara numerik, tetapi juga menjamin bahwa hasil tersebut dapat disampaikan secara efektif dan mudah dipahami dalam lingkungan virtual yang adaptif dan interaktif. Evaluasi ganda ini mencerminkan keseimbangan antara *computational performance* dan human-centered *usability*, yang menjadi inti dari pengembangan sistem cerdas berbasis VR untuk analisis dan mitigasi fenomena lingkungan perkotaan.

## **2.10 Kerangka Teori**

Kerangka teori merupakan gambaran pendekatan dalam penelitian visualisasi ruang virtual *urban* dengan memanfaatkan data citra satelit dan teknik pembelajaran mesin. Dalam Gambar 2.1 mengenai kerangka teori, citra satelit berfungsi sebagai sumber data utama (input) yang diproses melalui berbagai metode analisis, baik regresi maupun jaringan saraf tiruan, untuk menghasilkan visualisasi spasial *urban* yang lebih informatif dan dinamis. Proses ini

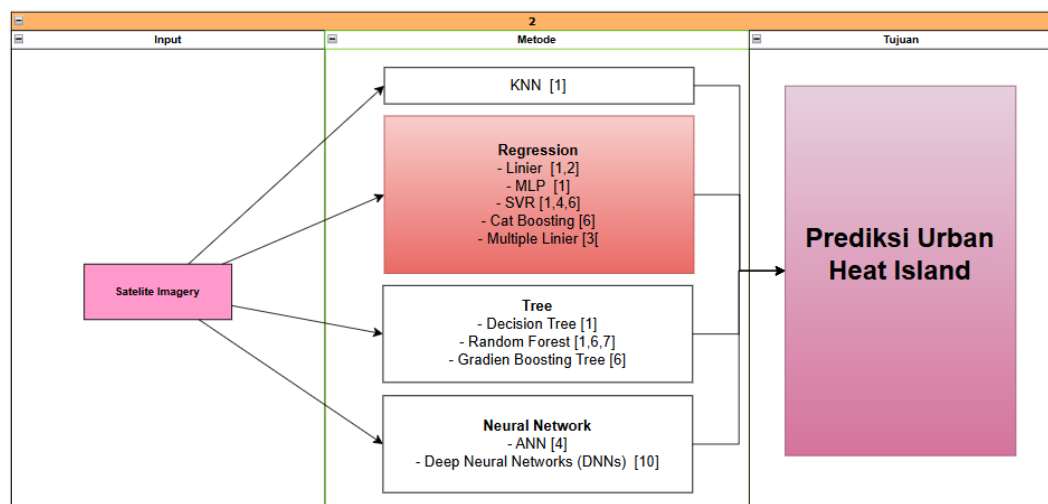
menekankan pada pemanfaatan data spasial multi-sumber Landsat, MODIS, dan Sentinel yang masing-masing memiliki karakteristik resolusi, frekuensi pengambilan data, dan cakupan wilayah yang berbeda-beda, memberikan keunggulan tersendiri dalam akurasi dan kedalaman informasi yang dihasilkan.

Dalam komponen proses, kerangka ini mengelompokkan metode analisis ke dalam dua pendekatan utama, yaitu regresi dan Neural Network. Pendekatan regresi terdiri atas beberapa metode seperti Regresi Linier (Zhao, 2022), *Random Forest* (Liu, 2023; Jato-Espino, 2022), Gradient Boosting (Fan, 2022), Regresi Non-Linier (Song, 2022), dan SVR (Farhan, 2024). Metode *Random Forest* dan Gradient Boosting mewakili algoritma pohon keputusan yang populer karena kemampuannya menangani data non-linear dan variabel dalam jumlah besar. SVR secara khusus menonjol karena kemampuannya dalam memetakan data ke ruang dimensi tinggi, yang sangat efektif dalam mengolah data kompleks seperti citra satelit multi-spektral. Sementara itu, Regresi Linier dan Non-Linier tetap relevan dalam konteks pemodelan awal dan analisis tren dasar.



Gambar 2. 1 Kerangka Teori

Di sisi lain, pendekatan Neural Network diwakili oleh ANN sebagaimana digunakan oleh Song (2022). ANN memiliki keunggulan dalam mengekstraksi pola non-linier dan kompleks dari data citra satelit, terlebih ketika data memiliki jumlah fitur yang besar dan hubungan antar-variabel yang tidak sederhana. Namun, pendekatan ini membutuhkan komputasi yang tinggi dan pelatihan data yang cermat untuk menghindari *overfitting*. Output akhir dari keseluruhan proses ini adalah Visualisasi Ruang Virtual *Urban*, yang memungkinkan representasi spasial yang lebih realistis dan mendalam terhadap kondisi lingkungan dan struktur *urban* berdasarkan hasil pemodelan dari berbagai metode tersebut. Dengan merujuk pada studi-studi terdahulu seperti Liu (2023), Jato-Espino (2022), Zhao (2022), Fan (2022), Song (2022), dan Farhan (2024), kerangka teori ini menjadi landasan konseptual yang kuat dalam menyusun sistem pemodelan *urban* berbasis citra satelit. Pemilihan metode analisis dilakukan berdasarkan keunggulan masing-masing dalam menangani karakteristik data input dan kebutuhan output visualisasi *urban* yang akurat dan efisien.



Gambar 2. 2 Kerangka Teori Prediksi UHI

Sebagai lanjutan dari kerangka teori, Gambar 2.2 menggambarkan alur pemodelan prediksi UHI yang dimulai dari pemanfaatan citra satelit sebagai *input*. Data citra satelit menyediakan variabel lingkungan seperti indeks vegetasi, indeks bangunan, dan suhu permukaan yang kemudian diproses menggunakan beberapa kelompok metode pembelajaran mesin. Tahap pemodelan dikelompokkan ke dalam empat pendekatan utama, yaitu KNN, regresi, *tree-based methods*, dan Neural Network. Metode regresi meliputi Regresi Linier, MLP, SVR, CatBoost, dan Regresi Multilinier, yang banyak digunakan untuk memetakan hubungan antara variabel lingkungan dan suhu permukaan. Pendekatan *tree-based* seperti Decision Tree, Random Forest, dan Gradient Boosting Tree digunakan karena kemampuannya menangani data non-linear dan interaksi variabel yang kompleks. Sementara itu, metode Neural Network, termasuk ANN dan DNN, berfungsi mempelajari pola non-linear yang lebih dalam dari data citra satelit. Seluruh metode ini diarahkan pada tujuan akhir yang ditunjukkan dalam kerangka, yaitu menghasilkan model prediksi UHI yang akurat. Dengan demikian, kerangka teori ini menekankan integrasi data citra satelit dan algoritma pembelajaran mesin sebagai fondasi dalam membangun sistem prediksi UHI sebelum masuk ke tahap visualisasi atau analisis lanjutan.

## **BAB III**

### **METODOLOGI PENELITIAN**

Dalam penelitian ini, serangkaian langkah telah dilakukan secara berurutan dan sistematis untuk memudahkan peneliti dalam menentukan langkah berikutnya, sehingga hasil yang diharapkan dapat tercapai

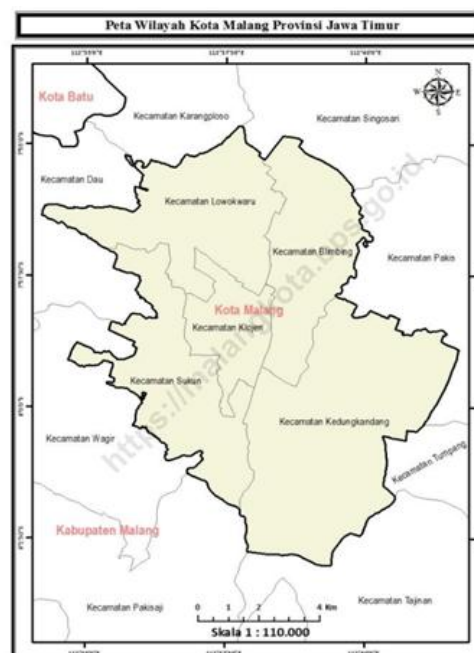
#### **3.1 Study Area**

Kota Malang merupakan salah satu kota metropolitan di Provinsi Jawa Timur yang memiliki laju *urbanisasi* dan pertumbuhan penduduk yang sangat pesat dalam dua dekade terakhir. Pertumbuhan kawasan permukiman, perdagangan, pendidikan, dan transportasi telah mendorong perubahan signifikan dalam struktur tata ruang kota. Konversi lahan vegetatif menjadi area terbangun, tingginya kepadatan bangunan, dan minimnya Ruang Terbuka Hijau (RTH) berkontribusi pada terbentuknya fenomena UHI, yakni kondisi suhu permukaan perkotaan yang lebih tinggi dibandingkan daerah sekitarnya yang tidak terbangun.

Secara administratif, Kota Malang memiliki luas wilayah sebesar 111,077 km<sup>2</sup> dan terbagi ke dalam lima kecamatan, yaitu: Kecamatan Blimbing, Kecamatan Klojen, Kecamatan Kedungkandang, Kecamatan Lowokwaru, dan Kecamatan Sukun. Masing-masing kecamatan menunjukkan karakteristik spasial dan penggunaan lahan yang berbeda-beda, yang berdampak terhadap pola distribusi suhu permukaan dan tingkat intensitas UHI. Pada Gambar 3.1 ditampilkan peta administratif Kota Malang yang menandai kelima kecamatan tersebut sebagai area studi.



Kecamatan dengan wilayah terluas adalah Kecamatan Kedungkandang, mencakup area sebesar 39,852 km<sup>2</sup> atau sekitar 35,88% dari total luas Kota Malang. Sebaliknya, kecamatan dengan wilayah terkecil adalah Kecamatan Klojen, yang hanya mencakup area seluas 8,829 km<sup>2</sup> atau sekitar 7,95% dari keseluruhan kota. Kota Malang sendiri berada pada ketinggian antara 445 hingga 526 meter di atas permukaan laut, dengan kontur topografi yang relatif bervariasi dan iklim tropis basah yang memengaruhi pola suhu dan kelembaban harian (Malang Dalam Angka, 2024).



Gambar 3. 1 Peta Administratif Kota Malang (Sumber : BPS)

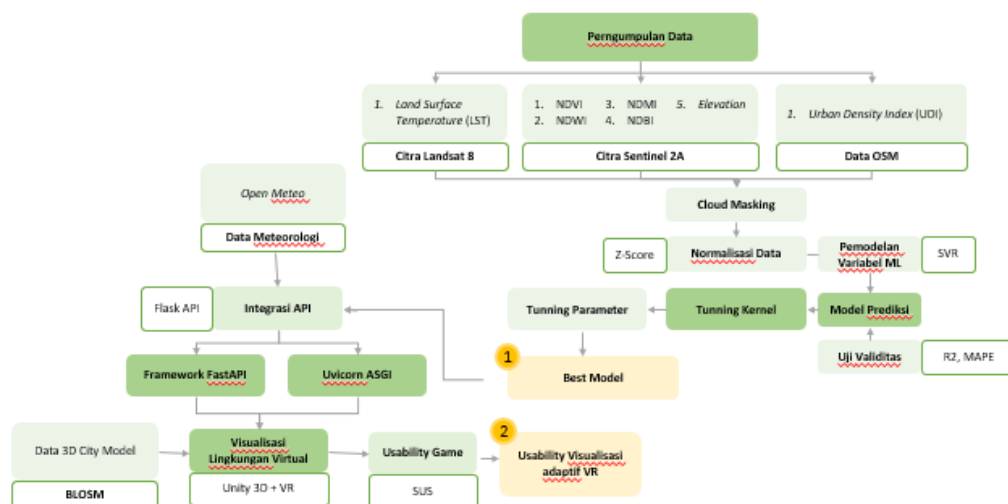
Kelima kecamatan tersebut dipilih sebagai area studi karena mewakili variasi densitas *urban*, distribusi ruang hijau, serta intensitas aktivitas antropogenik yang berbeda. Misalnya, Kecamatan Lowokwaru dan Blimbing dikenal sebagai kawasan pendidikan dan permukiman dengan tingkat kepadatan sedang hingga tinggi, sementara Kedungkandang masih memiliki sejumlah area terbuka dan

lahan pertanian yang mulai terkonversi. Kecamatan Klojen, meski paling kecil, merupakan pusat kota administratif dan memiliki kepadatan bangunan tertinggi.

Pemahaman terhadap karakteristik fisik dan spasial masing-masing kecamatan sangat penting dalam mengidentifikasi faktor-faktor dominan pembentuk UHI di Kota Malang. Dengan pendekatan ini, analisis dapat dilakukan secara lebih kontekstual dan presisi terhadap peran variabel lingkungan *urban* seperti albedo permukaan, jenis material bangunan, dan proporsi vegetasi terhadap fluktuasi suhu permukaan di tiap kecamatan.

### 3.2 Desain Penelitian

Penelitian ini menggabungkan pendekatan spasial, pemodelan prediktif, dan visualisasi virtual untuk menganalisis dan menyimulasikan kondisi termal lingkungan perkotaan.



Gambar 3. 2 Desain Penelitian

Gambar 3.2. Menunjukkan desain penelitian yang terdiri atas tiga tahapan utama, yaitu: pengumpulan data, desain sistem, serta evaluasi performa. Desain ini disusun secara sistematis untuk menjawab permasalahan penelitian mengenai kebutuhan prediksi suhu permukaan perkotaan yang akurat dan informatif serta kebutuhan akan visualisasi yang komunikatif dalam konteks mitigasi UHI.

Pada Tahap 1 yaitu tahap Pengumpulan Data, data dikumpulkan dari dua sumber utama, yaitu data satelit dan data geospasial tiga dimensi. Data penginderaan jauh satelit diperoleh melalui platform Google Earth Engine (GEE) dengan menggunakan tiga jenis citra utama: (1) *Landsat-8 OLI/TIRS Collection 2 Level-1 (Band 10)* untuk memperoleh Land Surface Temperature (LST) dengan skala 0.00341802 dan offset 149, sehingga memungkinkan estimasi suhu permukaan yang lebih detail. (2) *Sentinel-2 Level-2A (Band B3, B4, B8, B11, dan B12)* dengan skala 0.0001 untuk mengekstraksi lima variabel lingkungan. (3) SRTM untuk menghitung elevasi dengan resolusi 30 meter. Sebelum fitur dihitung, seluruh citra melalui proses pra-pemrosesan seperti *cloud masking*, *cropping*, *resampling*, serta normalisasi nilai reflektansi guna memastikan kualitas data yang baik dan konsisten. Di sisi lain, pembangunan model kota 3D dilakukan menggunakan data spasial dari *OpenStreetMap* (OSM) yang diolah melalui *plugin* Blender-GIS dan pemanfaatan GEE untuk ekstraksi data spasial dan lingkungan. Model kota 3D ini kemudian digunakan sebagai basis visualisasi yang akan diintegrasikan dengan hasil prediksi suhu dari model machine learning.

Tahap selanjutnya adalah Tahap 2 mulai masuk di Pengembangan Model Prediksi SVR. Pada tahap ini, dilakukan proses komputasi untuk ekstraksi fitur

dari data lingkungan, termasuk indeks vegetasi, indeks bangunan, indeks kelembapan, ketinggian, serta variabel geometris dari lingkungan terbangun. Data hasil ekstraksi fitur kemudian dinormalisasi menggunakan metode Z-Score untuk memastikan setiap variabel memiliki kontribusi yang setara dalam model prediktif. Selanjutnya dilakukan pemilihan parameter kernel dan proses tuning hyperparameter pada model SVR, yang diimplementasikan menggunakan bahasa pemrograman Python. Evaluasi performa model dilakukan dengan dua metrik utama, yaitu koefisien determinasi ( $R^2$ ) dan Mean Absolute Percentage Error (MAPE), untuk mengukur ketepatan model dalam memprediksi suhu permukaan. Model terbaik yang dihasilkan kemudian dikemas dalam layanan Application Programming Interface (API) menggunakan *framework* Flask untuk memungkinkan integrasi antara sistem pemodelan dan sistem visualisasi secara *real-time*.

Tahap berikutnya, Tahap 3, adalah *Integrasi API*. Model terbaik hasil dari proses pelatihan SVR dikemas menjadi layanan Application Programming Interface (API) menggunakan *framework* Flask, yang memungkinkan pertukaran data secara dinamis antara sistem pemodelan dan sistem visualisasi. API ini berfungsi sebagai jembatan komunikasi agar hasil prediksi suhu permukaan dapat diakses secara *real-time* oleh aplikasi visualisasi berbasis VR. Dengan demikian, sistem mampu menampilkan pembaruan data suhu terkini tanpa perlu pemrosesan ulang manual, sekaligus mendukung fleksibilitas integrasi dengan sumber data eksternal lainnya seperti Open Meteo untuk pengambilan data cuaca secara langsung.

Tahap terakhir, Tahap 4, adalah *Visualisasi Lingkungan Virtual*. Model kota 3D yang telah dibangun dan diperkaya dengan tekstur bangunan serta elemen visual lainnya diintegrasikan ke dalam platform visualisasi yang mendukung teknologi VR. Visualisasi ini bertujuan untuk menyajikan hasil prediksi suhu secara intuitif melalui perubahan warna permukaan, tekstur, dan elemen visual lainnya, sehingga mampu merepresentasikan fenomena UHI secara visual dan interaktif. Selain itu, dilakukan evaluasi *usability* menggunakan metode SUS terhadap sistem visualisasi yang dibangun untuk memastikan sistem tidak hanya informatif namun juga mudah digunakan dan relevan bagi pengguna akhir seperti pemerintah daerah, perencana tata kota, dan masyarakat umum. Sistem yang diusulkan ini dirancang untuk mendukung pengambilan keputusan berbasis data dan teknologi dalam konteks perencanaan kota berkelanjutan dan adaptasi terhadap perubahan iklim.

### **3.2.1 Pengumpulan Data**

Berdasarkan desain penelitian pada Gambar 3.2, data yang digunakan dalam penelitian ini berasal dari tiga sumber utama, yaitu data model kota tiga dimensi (3D City Model), data penginderaan jauh satelit yang diambil melalui platform Google Earth Engine (GEE), serta data cuaca *real-time* yang diperoleh dari layanan Open Meteo melalui integrasi Application Programming Interface (API). Ketiga jenis data ini dipilih karena bersifat saling melengkapi: data 3D Model digunakan untuk membangun lingkungan visualisasi, data satelit digunakan untuk menghasilkan fitur lingkungan dan suhu permukaan (LST), sedangkan data Open

Meteo digunakan untuk memperbarui data termal secara *real-time* sehingga sistem dapat menampilkan perubahan suhu aktual di Kota Malang.

#### 3.2.1.1 3D City Model

Data **3D City Model** diperoleh dari **OpenStreetMap (OSM)** yang menyediakan data spasial bangunan, jalan, vegetasi, dan elemen permukaan lainnya pada wilayah Kota Malang. Data tersebut diolah menggunakan **Blender-GIS Plugin**, yang berfungsi untuk mengubah data spasial menjadi representasi geometri 3D dengan mempertahankan sistem koordinat geospasial aslinya. Dalam tahap pra-pemrosesan, dilakukan beberapa langkah, yaitu: (1) Pembuatan model 3D kota menggunakan Blender, (2) Penentuan titik koordinat dan pemetaan suhu permukaan tanah menggunakan QGIS (Quantum GIS), serta (3) Integrasi model spasial dan data suhu ke dalam Unity 3D. Model 3D yang dihasilkan berfungsi sebagai basis visualisasi utama dalam lingkungan VR, di mana setiap elemen geospasial akan dikaitkan dengan hasil prediksi suhu permukaan dari model *machine learning*.

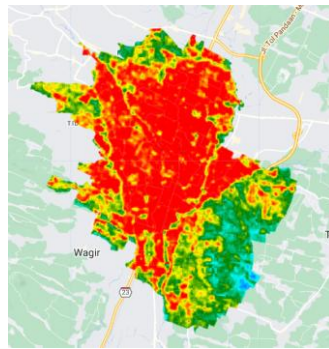
#### 3.2.1.2 Citra Satelit

Data **citra satelit** diperoleh melalui **Google Earth Engine (GEE)** yang memungkinkan akses, pengolahan, dan analisis data penginderaan jauh secara efisien berbasis komputasi awan (*cloud computing*). Tabel 3.1 menunjukkan tiga jenis citra utama digunakan dalam penelitian ini, yaitu **Landsat-8 OLI/TIRS Collection 2 Level-1**, **Sentinel-2 Level-2A**, dan **NASA SRTM**. Selain pengumpulan data, pada data citra satelit ini diperlukan ada preprocessing untuk menghasilkan citra yang akurat.

Tabel 3. 1 Citra Satelit yang akan digunakan

No	Citra	Band	Deskripsi	Scale	Offset
1.	Landsat-8 OLI/TIRS C2 L1	ST B10	LST	0.00341802	149
2.	Sentinel-2 Level-2A	B3	Green	0.0001	-
		B4	Red	0.0001	-
		B8	NIR	0.0001	-
		B11	SWIR 1	0.0001	-
		B12	SWIR 2	0.0001	-
3.	NASA SRTM Digital Elevation 30m	elevation	-	-	-

### 1. Landsat-8

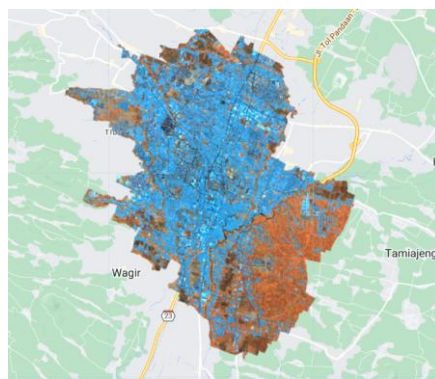


Gambar 3. 3 Citra Landsat 8 pada Band 10

Citra **Landsat-8** dipilih karena memiliki sensor termal *Thermal Infrared Sensor (TIRS)* yang dapat mendeteksi radiasi panas permukaan bumi, khususnya pada Band 10 dengan panjang gelombang 10,6–11,19  $\mu\text{m}$  yang berfungsi untuk menghitung *Land Surface Temperature (LST)* seperti pada Gambar 3.3. Resolusi spasialnya sebesar 100 meter kemudian di re-sampling menjadi 30 meter agar seragam dengan citra lainnya. Landsat-8 digunakan karena memiliki stabilitas radiometrik yang tinggi dan arsip temporal yang panjang, sehingga ideal untuk analisis suhu jangka menengah dan validasi model prediksi. Tahapan *preprocessing* mencakup *cloud masking* menggunakan *pixel\_qa band*, *cropping area* Kota Malang, serta kalibrasi radiometrik dengan faktor skala 0.00341802 dan

offset 149 untuk mengonversi nilai *Digital Number (DN)* menjadi *Top of Atmosphere Radiance (TOA)*. Nilai radiansi tersebut kemudian dikonversi menjadi suhu permukaan (Kelvin) menggunakan persamaan Planck dan diubah ke Celcius. Hasil akhir berupa peta distribusi suhu permukaan (LST) yang digunakan sebagai variabel target dalam model.

## 2. Sentinel-2 Level-2A



Gambar 3. 4 Gambar 3. 4 Citra Sentinel-2

Citra **Sentinel-2 Level-2A** digunakan untuk mengekstraksi fitur lingkungan yang memengaruhi suhu permukaan seperti vegetasi, kepadatan bangunan, kelembapan, dan air permukaan. Gambar 3.4 merupakan contoh citra sentinel-2. Sentinel-2 memiliki sensor **Multispectral Instrument (MSI)** dengan 13 kanal spektral dan resolusi tinggi (10–20 meter), sehingga mampu mengidentifikasi detail kecil di area perkotaan. Band yang digunakan meliputi B3 (Green, 560 nm), B4 (Red, 665 nm), B8 (NIR, 842 nm), B11 (SWIR1, 1610 nm), dan B12 (SWIR2, 2190 nm). Kombinasi band tersebut digunakan untuk menghitung indeks lingkungan utama seperti **NDVI (Normalized Difference Vegetation Index)** untuk vegetasi, **NDBI (Normalized Difference Built-up Index)** untuk area terbangun, **NDWI (Normalized Difference Water Index)** untuk air dan



kelembapan, **NDMI (Normalized Difference Moisturaize Index)** untuk air dan kelembapan, serta **UDI (Urban Density Index)** untuk kepadatan area perkotaan. Proses *preprocessing* meliputi *cloud masking* dengan algoritma QA60, koreksi atmosfer otomatis dari produk Level-2A untuk mengubah reflektansi *Top of Atmosphere* menjadi *Bottom of Atmosphere*, *resampling* ke resolusi 10 meter agar konsisten antarband, serta *normalisasi reflektansi* dengan skala 0.0001 agar nilai berada dalam rentang 0–1.

Setelah citra bersih dari pengaruh awan, tahapan berikutnya adalah **ekstraksi fitur**. Perhitungan dilakukan dengan rumus-rumus matematika yang sederhana untuk menghitung fitur-fitur yang akan digunakan. Fitur yang akan digunakan adalah NDVI, NDBI, NDWI, NDMI, UDI, elevasi, dan LST. Fitur NDVI, NDBI, NDWI, dan NDMI perlu dilakukan perhitungan berdasarkan Tabel 3. 2.

Tabel 3. 2 Fitur-fitur yang akan digunakan

No	Fitur	Keterangan	Perhitungan
1.	NDVI	Indeks yang digunakan dalam penginderaan jauh untuk mengukur tingkat kehijauan atau keberadaan vegetasi pada suatu area. Nilai tinggi menunjukkan adanya vegetasi yang subur.	$\frac{(NIR - Red)}{(NIR + Red)}$
2.	NDBI	Indeks yang digunakan dalam penginderaan jauh untuk mengukur area yang telah dibangun atau penggunaan lahan yang <i>terurbanisasi</i> .	$\frac{(SWIR2 - NIR)}{(SWIR2 + NIR)}$
3.	NDWI	Indeks yang digunakan dalam penginderaan jauh untuk mengukur kandungan air atau kelembaban dalam permukaan tanah	$\frac{(Green - NIR)}{(Green + NIR)}$
4.	NDMI	Indeks yang digunakan dalam penginderaan jauh untuk mengukur tingkat kelembaban atau keadaan hidrasi dari suatu wilayah	$\frac{(NIR - SWIR1)}{(NIR + SWIR1)}$
5.	UDI	Indeks yang digunakan dalam penginderaan jauh untuk mengukur tingkat <i>urbanisasi</i> .	-

6.	Elevasi	data elevasi permukaan bumi yang diukur oleh misi Shuttle Radar Topography Mission (SRTM). Memberikan informasi ketinggian suatu wilayah di atas permukaan laut.	-
7.	LST	Nilai suhu permukaan pada suatu daerah yang direkam menggunakan ST B10 dari satelit Landsat	-

#### a. NDVI

NDVI diperoleh dari rasio antara pantulan cahaya inframerah dekat (NIR) dan merah (Red) yang berasal dari vegetasi (Purwanto, 2015). Indeks ini digunakan untuk menilai kesehatan dan kerapatan vegetasi. Persamaan (3.1):

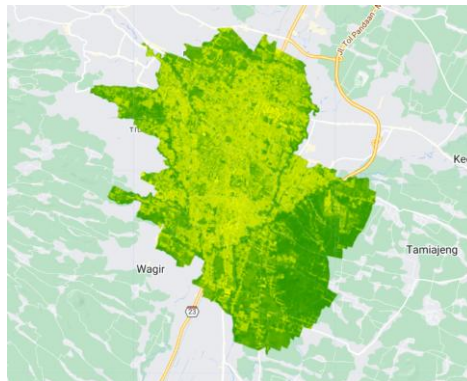
$$NDVI = \frac{(NIR - Red)}{(NIR + Red)} \quad (3.1)$$

Keterangan :

NIR : Band 8 Sentinel 2

RED : Band 4 Sentinel 2

\*merujuk pada tabel 3.1



Gambar 3. 5 NDVI

Berdasarkan Gambar 3. 5, rentang nilai NDVI berada pada rentang -1 hingga 1. Nilai mendekati 1 menunjukkan vegetasi yang lebat, sedangkan nilai negatif biasanya menunjukkan air, tanah gundul, atau permukaan non-vegetatif.

#### b. NDBI

Nilai NDBI merupakan bentuk gambaran dari tingkat kepadatan bangunan yang merupakan gambaran perkembangan perkotaan di suatu wilayah formulanya mirip dengan NDVI, tetapi NDBI membandingkan perbedaan antara intensitas inframerah dekat (NIR) dan SWIR2 dengan mengurangi nilai SWIR2 dari nilai inframerah dekat, kemudian membaginya dengan jumlah nilai SWIR2 dan inframerah dekat. Berikut ini adalah persamaan untuk menghitung nilai NDBI pada persamaan 3.2:

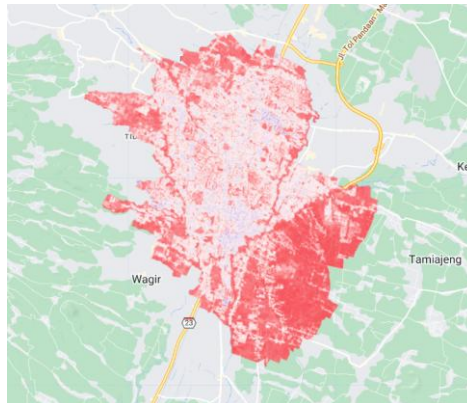
$$NDBI = \frac{(SWIR2 - NIR)}{(SWIR2 + NIR)} \quad (3.2)$$

Keterangan :

NIR : Band 8 Sentinel 2

SWIR2 : Band 12 Sentinel 2

\*merujuk pada tabel 3.1



Gambar 3. 6 NDBI

Berdasarkan Gambar 3. 6, rentang nilai NDBI berkisar antara -1 hingga 1. Nilai positif menunjukkan keberadaan material yang tidak memantulkan cahaya di spektrum merah dan menunjukkan kemungkinan area yang terbangun (bangunan atau infrastruktur). Sebaliknya, nilai negatif menunjukkan keberadaan material yang memantulkan cahaya di spektrum merah, seperti air atau vegetasi, dan menunjukkan kemungkinan area yang tidak terbangun.

### c. NDWI

Nilai NDWI digunakan untuk mendeteksi dan memetakan air pada citra penginderaan jauh. NDWI memanfaatkan perbedaan absorbansi antara dua panjang gelombang, yaitu inframerah dekat (NIR) dan hijau (G). NDWI dihitung dengan membandingkan jumlah cahaya hijau dan inframerah dekat (NIR) yang dipantulkan oleh badan air. Berikut ini adalah persamaan untuk menghitung nilai NDWI pada persamaan 3.3:

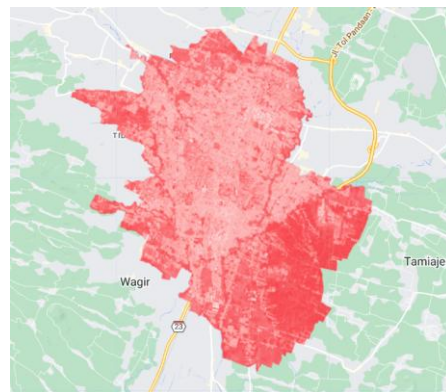
$$NDWI = \frac{(Green - NIR)}{(Green + NIR)} \quad (3.3)$$

Keterangan :

NIR : Band 8 Sentinel 2

Green : Band 3 Sentinel 2

\*merujuk pada tabel 3.1



Gambar 3. 7 NDWI

Berdasarkan Gambar 3. 7, rentang nilai NDWI berkisar antara -1 hingga 1. Area dengan nilai positif cenderung menunjukkan keberadaan air, sementara nilai negatif menunjukkan non-air. Nilai NDWI dapat membantu mengidentifikasi dan memetakan badan air seperti sungai, danau, atau wilayah tergenang.

### d. NDMI

Nilai NDMI digunakan untuk untuk menilai kelembaban atau kelembaban tanah pada citra penginderaan jauh. NDMI memanfaatkan perbedaan absorptansi antara dua panjang gelombang, yaitu inframerah dekat (NIR) dan SWIR1. Berikut ini adalah persamaan untuk menghitung nilai NDMI pada persamaan 3.4:

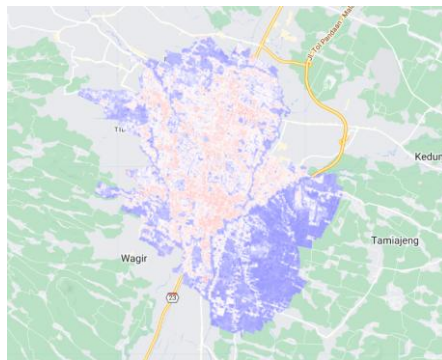
$$NDMI = \frac{(NIR - SWIR1)}{(NIR + SWIR1)} \quad (3.4)$$

Keterangan :

NIR : Band 8 Sentinel 2

SWIR1 : Band 11 Sentinel 2

\*merujuk pada tabel 3.1



Gambar 3. 8 NDMI

Berdasarkan Gambar 3. 8, rentang nilai NDMI berkisar antara -1 hingga 1. Nilai yang lebih tinggi cenderung menunjukkan kelembaban atau keberadaan air, sementara nilai yang lebih rendah dapat menunjukkan kekeringan atau kurangnya kelembaban. NDMI sering digunakan untuk memantau kondisi kelembaban tanah, mengidentifikasi wilayah yang memerlukan perhatian lebih dalam hal pengairan, seperti danau, sungai, dan lahan basah atau bahkan mendeteksi dampak kekeringan pada pertanian.

#### e. UDI

Indeks kepadatan kota (UDI) adalah ukuran yang digunakan dalam perencanaan kota dan desain kota untuk menilai jumlah orang yang mendiami suatu wilayah perkotaan. Hal ini merupakan faktor penting dalam memahami bagaimana kota berfungsi dan digunakan untuk mengkarakterisasi perluasan kota. UDI sering digunakan bersama dengan indeks lain, seperti indeks kepadatan kehijauan (GDI), untuk memberikan pemahaman yang lebih komprehensif tentang lingkungan perkotaan. GDI adalah indeks yang mengukur jumlah ruang hijau di dalam area perkotaan, dan dihitung dengan menggunakan indeks spektral yaitu NDWI, NDBI dan NBR2. Dari ketiga fitur tersebut lalu dihitung rata rata nilai pikselnya.

### 3. SRTM (Shuttle Radar Topography Mission)

Data **SRTM (Shuttle Radar Topography Mission)** digunakan untuk memperoleh informasi **elevasi** dengan resolusi spasial 30 meter. Elevasi menjadi variabel penting karena memengaruhi distribusi suhu dan pergerakan udara di wilayah dengan variasi topografi seperti Kota Malang. Data SRTM diunduh melalui GEE dan diproses melalui tahapan *void filling* untuk memperbaiki piksel kosong, *cropping* area penelitian, serta *reprojection* ke sistem koordinat WGS84 (EPSG:4326). Data elevasi ini kemudian dikombinasikan dengan citra Landsat dan Sentinel untuk memperkaya variabel input model prediksi suhu permukaan. Kombinasi dari ketiga citra satelit yaitu Landsat-8 untuk LST, Sentinel-2 untuk variabel lingkungan, dan SRTM untuk topografi menghasilkan dataset

multivariabel yang komprehensif, mencakup aspek termal, ekologis, dan geospasial dari wilayah perkotaan.

### 3.2.1.3 Open Meteo

Sumber data ketiga berasal dari Open Meteo, yaitu layanan API terbuka yang menyediakan data cuaca secara *real-time* dan historis berdasarkan koordinat geografis. Open Meteo dipilih karena mampu menyediakan data meteorologi secara gratis, akurat, dan terotomatisasi tanpa autentikasi pengguna, sehingga sangat sesuai untuk sistem integrasi berbasis *web service*. Data ini berfungsi sebagai sumber data dinamis untuk memperbarui informasi suhu dan kondisi atmosfer terkini yang akan dikirim ke sistem visualisasi VR melalui API berbasis Flask. Variabel utama yang digunakan meliputi temperature 2m, wind speed, relative humidity, dan evapotranspiration. Tahapan *preprocessing* dilakukan dengan menentukan grid koordinat di dalam batas administratif Kota Malang, kemudian melakukan pengambilan data secara berkala melalui permintaan *HTTP GET* ke endpoint API Open Meteo. Data cuaca yang diperoleh dalam format *JSON* dikonversi ke bentuk tabular dan disinkronkan secara temporal dengan data satelit menggunakan format waktu lokal (UTC+7).

Integrasi data Open Meteo dengan model SVR memungkinkan sistem melakukan **prediksi suhu permukaan secara adaptif dan *real-time***, di mana hasil prediksi langsung diteruskan ke sistem VR untuk divisualisasikan sebagai perubahan warna atau tekstur permukaan pada model kota 3D. Dengan demikian, kombinasi dari data 3D Model, citra satelit, dan data Open Meteo membentuk kerangka data yang utuh dan berlapis: mulai dari konteks spasial (3D), faktor

penyebab (satelit), hingga dinamika waktu (Open Meteo). Keseluruhan data ini menjadi fondasi utama bagi sistem visualisasi adaptif yang dikembangkan untuk menganalisis dan menampilkan fenomena UHI di Kota Malang secara komprehensif, akurat, dan interaktif.

### **3.2.2 Pengembangan Model Prediksi**

Tahapan pemodelan merupakan inti dari proses penelitian ini, di mana seluruh fitur hasil ekstraksi dari data satelit dan geospasial diolah menggunakan metode machine learning untuk memprediksi LST secara akurat. Algoritma yang digunakan dalam penelitian ini adalah SVR karena memiliki kemampuan tinggi dalam menangani data non-linear yang dibuktikan pada lampiran 1, menjaga keseimbangan antara kompleksitas model dan tingkat kesalahan, serta terbukti efektif dalam pemodelan data lingkungan berbasis penginderaan jauh. Tahapan pemodelan ini terdiri atas beberapa langkah utama, yaitu normalisasi data, pembagian data, pemilihan kernel dan tuning parameter, pembentukan fungsi regresi, evaluasi performa model, serta skenario pengujian. Masing-masing langkah dijelaskan secara rinci berikut ini.

#### **3.2.2.1 Normalisasi Data**

Langkah pertama dalam proses pemodelan adalah melakukan normalisasi data terhadap seluruh fitur yang telah diekstraksi dari hasil pengolahan citra satelit dan data spasial. Tujuan normalisasi adalah untuk menyamakan skala antarvariabel agar setiap fitur memiliki kontribusi yang seimbang dalam proses pelatihan model. Tanpa normalisasi, fitur dengan skala besar seperti elevasi



(dalam meter) dapat mendominasi variabel lain seperti NDVI yang memiliki rentang -1 hingga 1, sehingga memengaruhi kestabilan optimisasi model SVR.

Metode normalisasi yang digunakan dalam penelitian ini adalah Z-Score Normalization. Metode ini bekerja dengan cara mengubah nilai asli suatu variabel  $x$  ke dalam bentuk skor baku  $z$ , yang menggambarkan seberapa jauh nilai tersebut dari rata-rata dalam satuan simpangan baku. Persamaan umum normalisasi Z-Score dinyatakan sebagai:.

$$Normalisasi = \frac{x - \mu}{\sigma} \quad (3.5)$$

Keterangan :

$x$  = nilai variabel

$\mu$  = Rata-rata

$\sigma$  = Standar deviasi

Melalui transformasi ini, setiap fitur memiliki rata-rata 0 dan standar deviasi 1, sehingga model SVR dapat melakukan pembobotan dengan proporsional tanpa bias akibat perbedaan skala. Selain itu, normalisasi juga membantu meningkatkan efisiensi komputasi dan mempercepat proses konvergensi dalam optimisasi kernel. Dengan demikian, setiap variabel dapat dianalisis dan dibandingkan secara setara selama proses pelatihan. Proses normalisasi ini diterapkan pada seluruh dataset, baik data latih maupun data uji, menggunakan nilai rata-rata dan simpangan baku yang dihitung dari data latih.

### 3.2.2.2 Pembagian Data

Setelah proses normalisasi, langkah selanjutnya adalah **pembagian data** menjadi dua subset utama, yaitu data latih (*training set*) dan data uji (*testing set*). Data latih digunakan untuk membangun model prediksi, sedangkan data uji

digunakan untuk mengukur kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Pembagian dataset ini dilakukan dengan beberapa rasio berbeda untuk melihat pengaruh proporsi data terhadap hasil prediksi. Tiga skenario pembagian digunakan, yaitu 90%:10%, 80%:20%, dan 70%:30%. Pembagian dengan rasio besar pada data latih (90%) memungkinkan model belajar dengan lebih banyak data namun berisiko menurunkan kemampuan generalisasi, sedangkan pembagian dengan data uji lebih besar (30%) meningkatkan ketelitian evaluasi namun mengurangi jumlah sampel pembelajaran. Proses pembagian dilakukan secara acak menggunakan *random state* yang tetap untuk menjamin reproduisibilitas hasil eksperimen. Selain itu, dilakukan *stratified splitting* untuk memastikan distribusi fitur dan target tetap proporsional di kedua subset. Setiap skenario pembagian data ini kemudian dijalankan secara independen dengan prosedur tuning parameter dan evaluasi performa yang sama, sehingga dapat diperoleh skenario terbaik berdasarkan hasil akurasi dan stabilitas mode.

### 3.2.2.3 Metode SVR

Algoritma **SVR** merupakan pengembangan dari *Support Vector Machine* (SVM) yang digunakan untuk permasalahan regresi. SVR bekerja dengan mencari fungsi regresi ( $y$ ) yang dapat memprediksi nilai target dengan deviasi tidak lebih dari ambang batas kesalahan  $\epsilon$ , sekaligus meminimalkan kompleksitas model agar tidak terjadi *overfitting*. Konsep utama SVR adalah membangun hyperplane atau garis regresi di ruang berdimensi tinggi yang meminimalkan kesalahan prediksi di luar batas toleran. Fungsi regresi dalam SVR dapat dinyatakan sebagai:

$$y = (w x) + b \quad (3.6)$$

Keterangan:

$y$  = nilai prediksi  
 $w$  = bobot  
 $x$  = nilai variabel  
 $b$  = nilai bias

Dari persamaan 3.6, nilai  $y$  merupakan nilai prediksi yang ingin diprediksi oleh model,  $w$  (*weight*) adalah koefisien yang mengukur pengaruh masing-masing variabel atau fitur terhadap prediksi,  $x$  adalah vektor yang berisi variabel atau fitur-fitur yang digunakan untuk membuat prediksi, dan  $b$  adalah nilai bias atau offset yang memungkinkan model untuk melakukan penyesuaian terhadap prediksi. *Hyperplane* ini berperan sebagai pemisah antara data di atas dan di bawahnya, memungkinkan model untuk melakukan prediksi berdasarkan variabel input dengan memanfaatkan bobot ( $w$ ) dan bias ( $b$ ) yang telah diatur selama proses pelatihan model. Dalam kasus data non-linear,  $(wx)$  diimplementasikan melalui fungsi kernel. SVR dipilih dalam penelitian ini karena memiliki keunggulan dalam menangani data dengan hubungan non-linear yang kompleks,

seperti fenomena UHI yang dipengaruhi oleh banyak faktor lingkungan (vegetasi, kelembapan, elevasi, dan kepadatan bangunan) secara simultan dan tidak linier. Pelatihan model SVR melibatkan proses optimisasi fungsi objektif untuk meminimalkan nilai *regularized loss*.

#### **3.2.2.4 Pemilihan Kernel dan Tuning Hyperparameter**

Pemilihan kernel merupakan tahap krusial karena menentukan bagaimana data diproyeksikan ke ruang fitur berdimensi tinggi. Tiga jenis kernel yang digunakan dalam penelitian ini adalah Linear, Polynomial, dan RBF. Kernel linear digunakan untuk hubungan yang cenderung linier antar variabel; kernel polinomial memungkinkan pembentukan permukaan regresi yang lebih kompleks berdasarkan derajat polinomial. Sedangkan kernel RBF menjadi pilihan utama untuk menangkap hubungan non-linier yang sangat dinamis, seperti variasi suhu permukaan akibat distribusi vegetasi dan permukiman. Setiap kernel memiliki parameter yang perlu dioptimalkan agar model menghasilkan kinerja terbaik. Parameter Cost mengatur seberapa besar model memperbolehkan kesalahan prediksi; nilai besar membuat model lebih fokus pada pengurangan *error* namun berisiko *overfitting*. Parameter  $\epsilon$  (epsilon) menentukan lebar margin insensitif terhadap kesalahan kecil, sementara parameter  $\gamma$  (gamma) mengontrol luas pengaruh satu titik data pada kernel RBF. Nilai-nilai parameter terbaik diperoleh berdasarkan kombinasi metrik evaluasi yang dijelaskan pada subbab berikutnya.

### 3.2.2.5 Pembentukan Kernel, Matriks Hessian, dan Proses Sequential Learning

Dalam proses optimisasi SVR, setiap sampel data dikonversi ke ruang fitur melalui fungsi kernel. Hasil perhitungan kernel ini membentuk **Matriks Gram** atau **Matriks Hessian**, yang digunakan dalam formulasi optimisasi kuadratik (Quadratic Programming) untuk mencari nilai Lagrange Multiplier ( $\alpha_i$  dan  $\alpha_i^*$ ). Matriks Hessian mencerminkan hubungan antar sampel dan menentukan seberapa besar kontribusi masing-masing titik terhadap pembentukan fungsi regresi akhir. Selama proses pelatihan, dilakukan **Sequential Learning** di mana nilai  $\alpha_i$  dan  $\alpha_i^*$  diperbarui secara iteratif hingga mencapai kondisi optimal berdasarkan kriteria Karush-Kuhn-Tucker (KKT). Pada setiap iterasi, kesalahan prediksi dihitung untuk menilai apakah pasangan variabel perlu disesuaikan.

### 3.2.2.6 Evaluasi Performa Model

Dalam penelitian ini untuk mengukur performa model prediksi menggunakan Koefisien determinasi ( $R^2$ ) dan *Mean Absolute Percentage Error* (MAPE) sebagai evaluasi hasil akhir dari pengukuran metode prediksi.

#### 1. Koefisien determinasi ( $R^2$ )

Koefisien determinasi ( $R^2$ ) merupakan alat untuk mengukur besarnya keragaman variabel terikat atau variabel *dependent* yang mampu dijelaskan oleh variabel bebas atau variabel *independent* dalam model. Nilai  $R^2$  menunjukkan kebaikan model. Model yang baik memiliki nilai  $R^2$  yang mendekati angka 1, sedangkan model yang tidak baik memiliki nilai  $R^2$  mendekati angka 0. Kisaran

dari nilai nilai  $R^2$  yaitu dari 0 hingga 1 atau  $0 \leq R^2 \leq 1$  (Evy Sulistianingsih, 2018).

$$R^2 = 1 - \frac{RSS}{TSS} \quad (3.7)$$

Keterangan :

RSS = Jumlah kuadrat residual

TSS = Jumlah total kuadrat

## 2. Mean Absolute Percentage Error (MAPE)

MAPE menghitung persentase penyimpangan antara data aktual dengan data prediksi. Semakin kecil nilai MAPE, peramalan yang dilakukan semakin baik (F. Zhang & O'Donnell, 2019). MAPE dihitung menggunakan persamaan 3.8 sebagai berikut :

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \times 100\% \quad (3.8)$$

Keterangan :

$n$  = jumlah data

$A_t$  = nilai aktual pada periode  $t$

$F_t$  = nilai prediksi pada periode  $t$

Kriteria nilai MAPE untuk mengetahui standar minimal dari kinerja model prediksi disajikan pada Tabel 3.3.

Tabel 3. 3 Kriteria Nilai MAPE (F. Zhang & O'Donnell, 2019)

Nilai MAPE	Kriteria
<10%	Sangat Baik
10%-20%	Baik
20%-50%	Cukup
>50%	Buruk

### 3.2.2.7 Skenario Pengujian dan Pemilihan Model Terbaik

Tahap terakhir adalah menentukan **skenario pengujian** untuk membandingkan performa model pada berbagai konfigurasi data dan parameter. Setiap skenario pembagian data diuji dengan tiga jenis kernel dan kombinasi parameter yang berbeda. Dari setiap pengujian, nilai  $R^2$ , MAPE, dan MSE direkapitulasi dan dibandingkan. Skenario dengan performa paling stabil yakni kombinasi kernel dan parameter yang menghasilkan akurasi tinggi di semua proporsi data ditetapkan sebagai **model terbaik**. Hasil tuning dan konfigurasi model terbaik ini kemudian diserialisasi menggunakan format *pickle* (.pkl) agar dapat digunakan kembali pada tahap integrasi API, sehingga sistem dapat melakukan prediksi suhu permukaan secara *real-time* tanpa perlu melatih ulang model.

### 3.2.3 Tahap Integrasi API

Tahap integrasi API merupakan proses penghubung antara model prediksi yang telah dikembangkan dengan sistem visualisasi berbasis VR. Pada tahap ini, model terbaik hasil dari metode SVR dikemas dalam bentuk layanan Application Programming Interface (API) menggunakan *framework* Flask (Python). Tujuan utama dari tahap ini adalah agar sistem dapat mengakses hasil prediksi suhu permukaan secara dinamis dan *real-time* tanpa perlu melakukan proses pelatihan ulang. Arsitektur sistem API menggunakan konsep client-server, di mana server berperan sebagai pengelola model dan komputasi prediksi, sedangkan aplikasi visualisasi VR bertindak sebagai klien yang mengirimkan permintaan data dan menerima hasil prediksi. Data masukan yang dikirim melalui API berupa fitur

lingkungan hasil pengolahan citra satelit (NDVI, NDWI, NDBI, UDI, dan elevasi) serta data meteorologi *real-time* dari Open Meteo, meliputi suhu udara, kelembapan, kecepatan angin, dan radiasi matahari. Data tersebut kemudian dinormalisasi sesuai parameter yang digunakan saat pelatihan model, sebelum diproses oleh model SVR untuk menghasilkan prediksi suhu permukaan (LST).

Hasil prediksi dikirim kembali oleh API dalam format JavaScript Object Notation (JSON) yang berisi nilai suhu permukaan beserta waktu dan koordinat pengamatan. Nilai ini selanjutnya digunakan oleh sistem visualisasi untuk menampilkan perubahan warna permukaan kota sesuai dengan tingkat suhu yang diprediksi. Melalui mekanisme ini, integrasi antara API dan sistem VR memungkinkan visualisasi fenomena UHI dilakukan secara interaktif, adaptif, dan berbasis data *real-time*. Untuk menjamin keandalan sistem, dilakukan pengujian fungsi API meliputi aspek akurasi keluaran, waktu respon, dan stabilitas komunikasi. Hasil evaluasi menunjukkan bahwa API dapat bekerja secara efisien dalam mengirim dan menerima data, serta mempertahankan hasil prediksi yang konsisten dengan model SVR yang telah disimpan. Tahapan ini menjadi dasar bagi proses visualisasi adaptif pada tahap selanjutnya, di mana nilai prediksi yang dihasilkan akan direpresentasikan secara visual dalam bentuk peta termal 3D Kota Malang.

#### **3.2.4 Visualisasi Adaptif dalam Lingkungan VR**

Tahap visualisasi merupakan fase akhir dari penelitian ini yang bertujuan untuk menyajikan hasil prediksi suhu permukaan dari model SVR dalam bentuk representasi tiga dimensi yang interaktif menggunakan teknologi VR. Tahapan ini

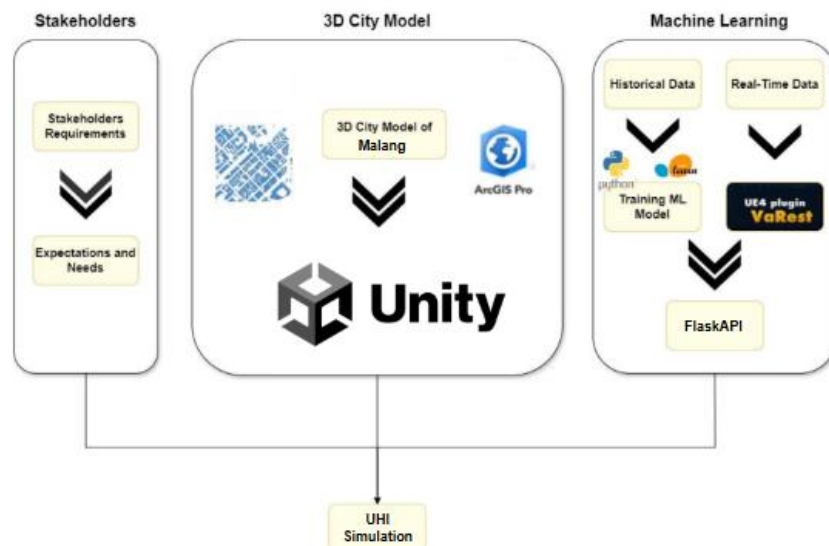


berfungsi sebagai bentuk konkret dari sistem visualisasi adaptif yang memungkinkan pengguna memahami fenomena UHI secara lebih intuitif, edukatif, dan kontekstual. Pendekatan VR dipilih karena kemampuannya menghadirkan lingkungan digital yang imersif, di mana pengguna dapat berinteraksi langsung dengan data spasial dan mengalami visualisasi perubahan suhu seolah berada di dalam kota sebenarnya. Dengan integrasi API yang menghubungkan hasil prediksi suhu secara *real-time*, sistem ini menjadi sarana pengamatan fenomena termal perkotaan yang adaptif terhadap data dinamis, sekaligus media pembelajaran interaktif bagi pengguna. Tahap ini terdiri atas lima komponen utama, yaitu (1) perancangan sistem dan alur visualisasi, (2) penerapan Finite State Machine (FSM) sebagai pengendali interaksi pengguna, (3) perancangan antarmuka pengguna (User Interface / UI), (4) integrasi hasil prediksi API ke dalam lingkungan VR, dan (5) evaluasi *usability* sistem menggunakan metode SUS.

#### 3.2.4.1. Desain Sistem

Desain sistem dalam penelitian ini dirancang untuk mengintegrasikan berbagai komponen yang bekerja secara terpadu mulai dari **analisis kebutuhan stakeholder**, **pembangunan model kota tiga dimensi (3D City Model)**, **pemrosesan data lingkungan menggunakan algoritma machine learning**, hingga **visualisasi adaptif berbasis Unity 3D dengan dukungan VR**. Sistem ini dikembangkan untuk menampilkan fenomena UHI secara interaktif dan *real-time*, sekaligus menjadi sarana pembelajaran dan analisis berbasis data ilmiah. Secara umum, arsitektur sistem digambarkan pada **Gambar 3.9**, yang menunjukkan tiga

pilar utama dalam desain sistem ini, yaitu *Stakeholders*, **3D City Model**, dan **Machine Learning**, yang saling terhubung melalui platform **Unity** sebagai pusat integrasi visualisasi dan interaksi.



Gambar 3. 9 Desain Sistem

Pada gambar 3.9 desain sistem, Tahapan awal dimulai dari identifikasi kebutuhan dan ekspektasi *stakeholder*, seperti pemerintah kota, perencana wilayah, dan masyarakat umum. Informasi tersebut menjadi dasar dalam merancang sistem visualisasi yang relevan dan aplikatif. Selanjutnya, dilakukan pembangunan model kota 3D Malang menggunakan data spasial yang diolah melalui perangkat lunak GIS. Di sisi lain, dilakukan pengumpulan dan pemrosesan data historis serta *real-time* pada google earth engine. Data ini digunakan untuk melatih model prediksi suhu menggunakan algoritma *SVR* yang dikembangkan dalam Python untuk dilakukan prediksi. Hasil prediksi suhu kemudian disimpan pada layanan API menggunakan Flask, yang memungkinkan komunikasi langsung antara model machine learning dan sistem visualisasi.

Model 3D yang telah dibangun kemudian diimpor ke dalam platform Unity untuk dilakukan proses visualisasi. Di dalam Unity, data prediksi suhu yang diterima dari API divisualisasikan secara adaptif melalui gradasi warna, perubahan tekstur, atau elemen visual lainnya yang menggambarkan fenomena UHI secara *real-time* maupun simulatif. Seluruh sistem dirancang untuk dapat diakses dalam bentuk VR, memungkinkan pengguna mengeksplorasi kondisi lingkungan secara imersif dan intuitif. Dengan pendekatan ini, sistem tidak hanya memberikan representasi visual kondisi termal kota, tetapi juga menjadi alat bantu pengambilan keputusan berbasis data dan teknologi, terutama dalam konteks mitigasi UHI di wilayah perkotaan seperti Kota Malang.

#### 1. *Stakeholder Analysis* untuk Perancangan Game

Tahapan awal desain sistem dimulai dengan *stakeholder analysis*, yang bertujuan untuk mengidentifikasi kebutuhan, ekspektasi, dan tantangan yang dihadapi oleh pihak-pihak terkait terhadap permasalahan fenomena UHI. *Stakeholder* yang dimaksud meliputi:

1. Pemerintah daerah, yang memerlukan sistem ini sebagai alat bantu pengambilan keputusan dalam perencanaan tata ruang dan mitigasi iklim mikro perkotaan.
2. Perencana kota dan arsitek lanskap, yang membutuhkan visualisasi spasial yang realistis untuk mengevaluasi pengaruh elemen vegetasi, permukaan, dan bangunan terhadap distribusi panas.

3. Peneliti dan akademisi, yang dapat memanfaatkan sistem ini sebagai media penelitian multidisiplin antara lingkungan, teknologi, dan data science.
4. Masyarakat umum dan pelajar, yang menjadi sasaran edukatif untuk meningkatkan kesadaran terhadap isu iklim perkotaan dan pentingnya ruang hijau.

Analisis kebutuhan ini menghasilkan dokumen spesifikasi sistem yang berisi daftar kebutuhan fungsional seperti simulasi perubahan suhu, pemantauan *real-time*, dan visualisasi VR dan non-fungsional seperti kinerja sistem, aksesibilitas, dan kenyamanan pengguna. Informasi ini menjadi dasar perancangan arsitektur data, rancangan antarmuka, dan skenario simulasi dalam game edukatif yang akan dikembangkan

## **2. Kebutuhan *Stakeholder* dan Komponen Pendukung Sistem**

Game edukatif ini dirancang sebagai sebuah simulasi interaktif yang bertujuan untuk memvisualisasikan fenomena UHI di Kota Malang secara adaptif, menggunakan pendekatan machine learning berbasis algoritma SVR. Game ini dibangun dengan engine Unity 3D dan mengintegrasikan model kota 3D hasil olahan QGIS, yang merepresentasikan elemen-elemen spasial kota seperti bangunan, vegetasi, dan jalan secara realistis. Pengguna berperan sebagai penjelajah lingkungan virtual yang dapat mengamati, mengevaluasi, serta memodifikasi lingkungan sekitar dan langsung melihat dampaknya terhadap kondisi suhu permukaan kota.

Melalui antarmuka first-person atau dukungan perangkat VR, pengguna dapat berkeliling kota dan melihat perbedaan suhu permukaan yang ditampilkan secara visual dengan gradasi warna. Data suhu ini berasal dari prediksi SVR yang dihasilkan dari data historis dan *real-time*, kemudian dikirim melalui Flask API ke Unity untuk divisualisasikan secara dinamis. Dalam mode simulasi, pengguna diberikan opsi untuk menambahkan elemen seperti vegetasi, mengubah permukaan tanah, atau menghapus bangunan tertentu. Setiap perubahan akan memicu proses analisis ulang oleh model SVR untuk memprediksi ulang suhu berdasarkan skenario baru yang kemudian ditampilkan secara langsung dalam visualisasi.

Sistem ini tidak hanya menyajikan eksplorasi visual, tetapi juga menyediakan fitur analitik berupa grafik atau panel informasi kecil yang menunjukkan perbandingan suhu sebelum dan sesudah modifikasi lingkungan. Tujuan dari desain ini adalah memberikan pengalaman yang edukatif, imersif, dan berbasis data, yang tidak hanya meningkatkan pemahaman pengguna terhadap fenomena UHI, tetapi juga memberikan gambaran dampak kebijakan atau perubahan tata ruang kota secara langsung. Game ini diharapkan dapat digunakan oleh *stakeholder* seperti pemerintah daerah, perencana tata ruang, pelajar, maupun masyarakat umum sebagai media edukasi dan pengambilan keputusan berbasis simulasi.

Beberapa kebutuhan utama stakeholder yang diidentifikasi dalam studi ini untuk mengembangkan sistem visualisasi UHI adalah sebagai berikut:

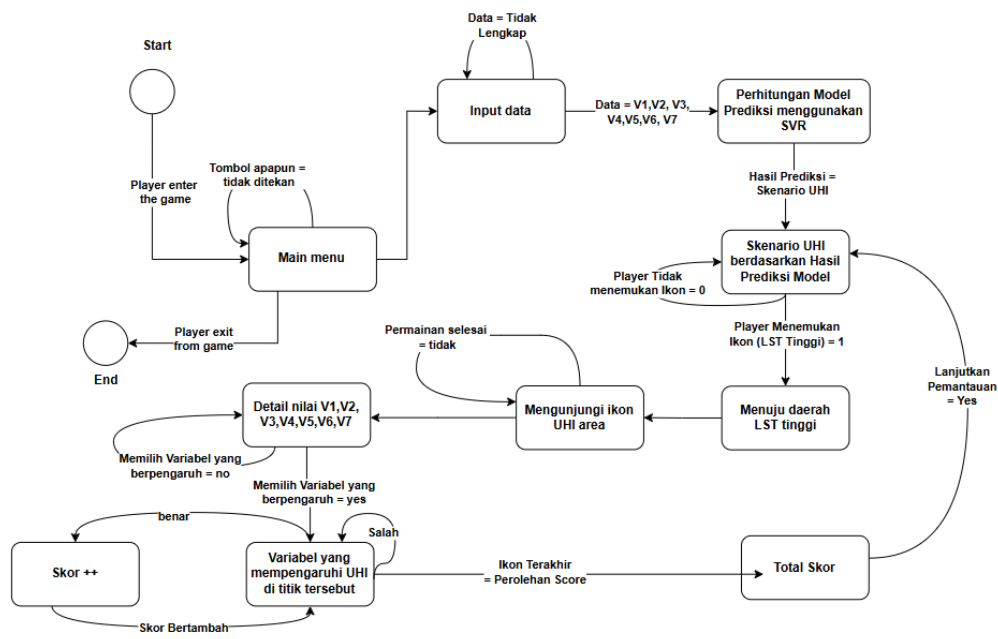
- Model Kota 3D sebagai Representasi visual dari wilayah kota secara tiga dimensi yang mencakup bangunan, jalan, vegetasi, air, kontur permukaan, dan elemen fisik lainnya. Model ini dikembangkan untuk menyediakan dasar representasi spasial.
- Data Prediksi Spasial : Sistem ini juga terhubung dengan GEE melalui API untuk memperoleh pembaruan data satelit secara berkala dan otomatis. Data prediksi dapat langsung digunakan untuk memperbarui visualisasi dan hasil prediksi. Perencana kota dan arsitek lanskap, yang membutuhkan visualisasi spasial yang realistis untuk mengevaluasi pengaruh elemen vegetasi, permukaan, dan bangunan terhadap distribusi panas.
- Data Meteorologi: Informasi cuaca seperti arah dan kecepatan angin juga digunakan untuk memperhitungkan faktor lingkungan yang berpengaruh pada distribusi panas di lingkungan kota.

Platform utama yang digunakan dalam pengembangan sistem ini adalah Unity karena kemampuannya dalam memvisualisasikan data 3D, mengintegrasikan data secara *real-time*, serta mendukung antarmuka interaktif. Unity terhubung dengan server Flask yang menyimpan model SVR, sehingga sistem ini memungkinkan pengguna untuk mengubah parameter dan melihat dampaknya terhadap prediksi LST dan UHI secara langsung.

### **3. Implementasi Sistem**

Sistem visualisasi adaptif kondisi lingkungan virtual dalam simulasi UHI yang dikembangkan dalam penelitian ini memanfaatkan metode SVR untuk

memprediksi suhu permukaan berdasarkan berbagai variabel lingkungan. Pada bagian ini, akan dijelaskan secara umum bagaimana sistem beroperasi dengan menggunakan diagram alur dan diagram urutan untuk memberikan gambaran yang lebih rinci mengenai proses prediksi suhu permukaan dan visualisasi adaptif kondisi lingkungan virtual. Alur proses ini dapat dilihat pada diagram alur yang ditunjukkan pada Gambar 3.10.



Gambar 3. 10 Finite State Machine

Perancangan sistem visualisasi adaptif berbasis gamifikasi dalam simulasi fenomena UHI mengacu pada pendekatan Finite State Machine (FSM) yang tersaji pada Gambar 3.10. FSM ini merepresentasikan alur interaktif antara pengguna (pemain) dan sistem berbasis model prediksi *SVR*, dengan tujuan untuk menyajikan simulasi edukatif yang menggabungkan unsur pembelajaran berbasis data spasial dan lingkungan. Pada awal permainan, sistem berada pada state Main

Menu, yang berfungsi sebagai antarmuka utama bagi pengguna. Pemain dapat memulai permainan dengan menekan tombol *Enter* atau tombol lain yang disediakan. Jika pemain tidak melakukan interaksi, sistem tetap berada di *Main Menu State* hingga tindakan terjadi. Setelah pemain menekan tombol untuk memulai, sistem berpindah ke Input Data State.

Pada state Input Data, sistem melakukan proses pemeriksaan kelengkapan data masukan (V1–V7), yang berisi variabel hasil ekstraksi dari model *SVR*, seperti NDVI, NDWI, NDBI, UDI, Elevasi, serta parameter meteorologi seperti kelembapan dan suhu udara. Bila data belum lengkap, sistem mengembalikan pesan kesalahan dan menunggu hingga data siap. Namun, jika data dinyatakan valid, maka sistem menjalankan Perhitungan Model Prediksi *SVR* untuk menghitung hasil prediksi suhu permukaan (*LST*) pada tiap titik koordinat di dalam kota.

Hasil dari perhitungan ini menghasilkan *Skenario UHI*, yaitu peta distribusi suhu perkotaan yang merepresentasikan tingkat intensitas UHI di berbagai wilayah. FSM kemudian berpindah ke state *Skenario UHI* berdasarkan Hasil Prediksi Model, yang menjadi dasar visualisasi permainan. Pada tahap ini, sistem mengatur posisi ikon-ikon UHI di area dengan nilai *LST* tertinggi, yang akan menjadi target interaksi pemain. Selanjutnya, permainan masuk ke state *Eksplorasi* atau *Mengunjungi Ikon UHI Area*. Di sini, pemain menjelajahi kota virtual dalam tampilan 3D untuk mencari dan mengunjungi ikon yang mewakili area dengan suhu permukaan tinggi. Saat pemain berhasil mendekati ikon UHI pertama (Player menemukan ikon = 1), sistem beralih ke state berikutnya yaitu



Detail Nilai Variabel (V1–V7). Dalam state Detail Variable, sistem menampilkan panel informasi yang berisi nilai-nilai setiap variabel lingkungan (NDVI, NDMI, NDBI, NDWI, UDI, dan Elevasi) pada titik yang sedang diamati. Pemain bisa menganalisis variabel mana yang paling berpengaruh terhadap peningkatan suhu pada area tersebut. Tahapan ini bertujuan untuk melatih pemahaman pengguna terhadap faktor penyebab UHI, sekaligus menjadi elemen edukatif dalam game.

Setelah pemain memilih salah satu variabel, FSM memasuki state Decision State. Pada state ini, sistem membandingkan jawaban pemain dengan nilai referensi hasil analisis model SVR. Setelah skor diperbarui, FSM akan kembali ke state Mengunjungi Ikon UHI Area untuk melanjutkan eksplorasi ke area berikutnya. Proses ini berlangsung secara berulang hingga seluruh ikon UHI pada peta telah dikunjungi. Apabila pemain berhasil menemukan semua ikon (ikon terakhir = true), sistem akan berpindah ke state Total Skor. Pada tahap ini, sistem menghitung total skor akhir berdasarkan akurasi jawaban pemain, jumlah variabel yang diidentifikasi dengan benar, serta waktu penyelesaian permainan. Nilai akhir ini kemudian ditampilkan sebagai hasil akhir permainan sebelum pemain kembali ke *Main Menu* atau keluar dari permainan (*End State*). FSM ini memastikan bahwa setiap tahap permainan memiliki alur transisi yang jelas dan tidak saling tumpang tindih. Dengan demikian, sistem dapat menjalankan proses pembelajaran interaktif yang berkelanjutan dan tetap stabil selama eksekusi.

#### **4. Desain Antarmuka (User Interface)**

Sistem dimulai dari kondisi *Start*, di mana pengguna diarahkan ke menu utama seperti yang ditunjukkan pada Gambar 3.11 Tampilan awal menampilkan

judul *Malang City* dengan tiga opsi utama: *New Game*, *Settings*, dan *Quit*. Menu ini berfungsi sebagai *state awal* dalam mekanisme Finite State Machine (FSM), di mana sistem menunggu input pengguna untuk menentukan jalur eksekusi berikutnya.



Gambar 3. 11 UI Main Menu

Ketika pemain memilih *New Game*, Tahap selanjutnya adalah sistem langsung melakukan fetching data. sistem secara otomatis melanjutkan ke Fetching Data State, di mana data lingkungan diambil dan diproses menggunakan SVR untuk menghasilkan nilai prediksi LST. Setelah prediksi LST berhasil diperoleh, sistem mengklasifikasikan hasilnya ke dalam lima kelas skenario UHI berurutan berdasarkan nilai rata-rata dan standar deviasi seperti pada tabel 3.4.

Tabel 3. 4 Klasifikasi kelas UHI (Nainggolan *et al.*, 2020)

Skenario Kelas UHI	Kelas Intensitas UHI	Rentang Intensitas UHI
1.	Dingin	$LST < \mu - 1.5\sigma$
2.	Sejuk	$\mu - 1.5\sigma < LST < \mu - 0.5\sigma$
3.	Sedang	$\mu - 0.5\sigma < LST < \mu + 0.5\sigma$
4.	Panas	$\mu + 0.5\sigma < LST < \mu + 1.5\sigma$
5.	Sangat Panas	$LST > \mu + 1.5\sigma$



Gambar 3. 11 Skenario UHI 1 (Dingin)



Gambar 3. 12 Skenario UHI 2 (Sejuk)



Gambar 3. 123 Skenario UHI 3 (Sedang)



Gambar 3. 14 Skenario UHI 4 (Panas)



Gambar 3. 15 Skenario UHI 5 (Sangat Panas)

Hasil prediksi LST kemudian diklasifikasikan ke dalam lima kelas skenario UHI seperti pada tabel 3.4 berdasarkan pendekatan  $\mu$  dan  $\sigma$ , mengadopsi metode dari Nainggolan et al. (2020). Pendekatan ini memungkinkan penentuan kategori intensitas panas secara objektif, berdasarkan penyebaran data aktual dibanding sekadar batas numerik absolut. Dengan demikian, setiap kelas bukan hanya representasi warna, tetapi cerminan dinamika distribusi panas yang terjadi secara spasial.

Pada Gambar 3.11 visualisasi skenario pertama, kategori “dingin,” lingkungan divisualisasikan dengan nuansa atmosfer yang sejuk dan stabil. Warna biru mendominasi permukaan tanah hingga horizon, menciptakan persepsi

kenyamanan termal. Awan berwarna putih cerah yang mengambang di langit biru menguatkan karakteristik rendahnya radiasi panas. Bangunan dan atap-atap rumah terlihat normal tanpa distorsi warna yang biasanya menandakan refleksi panas. Efek kabut hampir tidak terlihat, menunjukkan minimnya akumulasi panas di lapisan udara permukaan. Representasi visual ini menjadi baseline penting karena skenario dingin merepresentasikan area dengan nilai LST yang jauh di bawah rata-rata ( $LST < \mu - 1.5\sigma$ ), sehingga memberikan pemain titik referensi dasar tentang kondisi wilayah tanpa tekanan termal. Pada tahap ini, pemain diperlihatkan bagaimana lingkungan fisik terlihat ketika energi panas terserap secara optimal oleh vegetasi dan albedo permukaan rendah.

Pada Gambar 3.12 menunjukkan skenario kedua, kategori “sejuk,” lingkungan menampilkan perubahan atmosfer yang lebih subtle namun signifikan. Pada horizon, gradasi warna hangat mulai muncul menggantikan biru dominan. Kabut tipis muncul perlahan, memberikan kesan bahwa radiasi panas mulai terperangkap di lapisan bawah atmosfer. Permukiman terlihat masih stabil namun terdapat bias warna kekuningan yang menunjukkan meningkatnya refleksi panas permukaan. Warna langit yang awalnya cerah mulai menunjukkan tanda-tanda transisi termal, menggambarkan kondisi perkotaan yang mulai mengalami peningkatan suhu permukaan namun tetap berada dalam rentang aman. Kategori ini sesuai dengan LST antara  $\mu - 1.5\sigma$  dan  $\mu - 0.5\sigma$ , yang menunjukkan bahwa meskipun suhu lebih hangat dibanding skenario pertama, kondisi ini masih belum memberikan tekanan termal berlebih. Perubahan visual ini penting untuk

membantu pemain memahami titik awal transisi menuju kondisi UHI yang lebih intens.

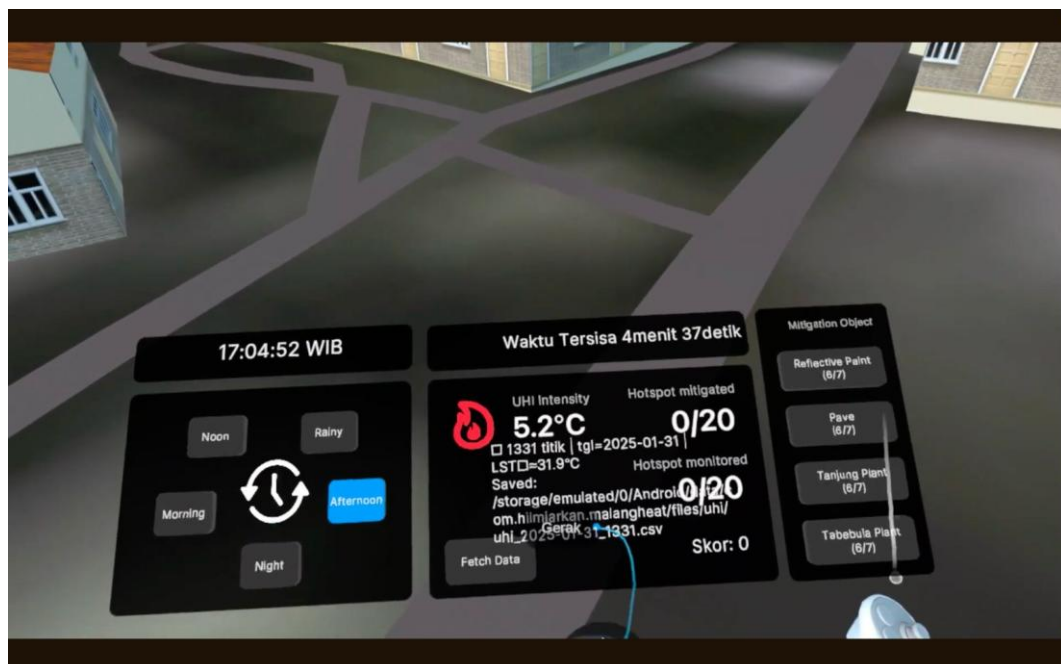
Gambar 3.13 menunjukkan skenario ketiga, kategori “sedang,” memperlihatkan visualisasi yang jauh lebih dinamis. Pada gambar ini, sistem mulai menampilkan panel interaktif yang menunjukkan nilai LST aktual dan prediksi yang dihasilkan model SVR untuk titik tertentu. Panel ini mencakup informasi koordinat, nilai LST prediksi, nilai LST baseline, serta variabel pendukung seperti NDVI, albedo, dan densitas bangunan. Lingkungan sekitar tampak netral, tidak terlalu dingin atau terlalu panas, dengan gradasi warna udara yang menunjukkan mulai adanya akumulasi panas yang lebih signifikan. Efek heat shimmer halus terlihat pada permukaan atap rumah sebagai indikasi reflektansi tinggi. Dominasi warna oranye lembut pada horizon menandakan kenaikan energi termal yang lebih merata. Skenario ini menggambarkan kondisi umum perkotaan saat siang hari, sesuai rentang  $\mu - 0.5\sigma < LST < \mu + 0.5\sigma$ . Pada tahap ini, pemain diperkenalkan pada kondisi yang paling sering terjadi di lapangan, sehingga membantu membangun pemahaman intuitif bahwa intensitas panas perkotaan secara alami cenderung berkisar pada zona sedang ini sebelum meningkat menjadi UHI.

Pada Gambar 3.14 menunjukkan skenario keempat, kategori “panas,” intensitas visual meningkat signifikan. Lingkungan berubah lebih drastis dengan dominasi warna oranye kemerahan yang lebih pekat. Permukaan bangunan terlihat dipenuhi pantulan cahaya terang, menciptakan kesan radiasi panas yang kuat. Kabut semakin tebal, memberikan efek atmosfer yang menandakan energi panas

terperangkap di lapisan udara dekat permukaan, sebuah kondisi yang menjadi ciri fenomena UHI menengah–tinggi. Panel interaktif kembali muncul, menampilkan nilai LST yang semakin tinggi serta indikator variabel lingkungan yang menunjukkan kontribusi faktor penyebab seperti minimnya vegetasi dan kepadatan bangunan yang ekstrem. Skenario ini sesuai dengan rentang  $\mu + 0.5\sigma < LST < \mu + 1.5\sigma$ . Kombinasi visual, data panel, dan penambahan efek atmosfer memperkuat pesan bahwa kondisi lingkungan pada kategori ini sudah mulai berbahaya bagi kenyamanan termal penduduk.

Memasuki skenario kelima yang ditunjukkan Gambar 3.15, kategori “sangat panas,” intensitas visual mencapai titik ekstrem. Lingkungan VR berubah drastis dengan dominasi warna merah menyala yang memancar pada seluruh permukaan. Atmosfer tampak berlapis kabut panas padat dengan efek distorsi visual yang mencerminkan radiasi panas intens. Panel interaktif kini menampilkan nilai LST yang jauh melampaui rata-rata, mengonfirmasi bahwa titik ini termasuk hotspot UHI paling kritis. Efek visual berupa awan merah menggantung tepat di atas lokasi terpanas, memberikan simbol kuat bahwa area tersebut merupakan pusat intensitas panas tinggi yang perlu mendapat perhatian intens dalam skenario mitigasi. Skenario ini sesuai dengan rentang  $LST > \mu + 1.5\sigma$ . Visualisasi ekstrem ini tidak hanya berfungsi sebagai peringatan visual, tetapi juga memperkuat pemahaman pemain tentang urgensi mitigasi UHI di daerah tertentu. Perpaduan warna merah, kabut panas, dan distorsi udara membuat pemain secara intuitif dapat merasakan tekanan termal bahkan tanpa membaca data panel.

Selanjutnya sistem melanjutkan proses ke tahap pembangkitan hotspot. FSM otomatis memilih 100 titik terpanas di seluruh wilayah kota berdasarkan ranking nilai LST. Setiap titik ditandai dengan visual indikator dan dapat diklik pemain untuk menampilkan informasi lengkap. Pada tahap ini, pemain dapat melakukan analisis mikro terhadap pola spasial UHI, membandingkan nilai prediksi antar titik, dan memahami perbedaan karakteristik lingkungan yang menyebabkan titik tertentu menjadi hotspot. Secara keseluruhan, kombinasi model prediktif, klasifikasi berbasis statistik, dan visualisasi VR yang imersif dalam lima skenario UHI ini memberikan pengalaman yang tidak hanya edukatif tetapi juga informatif. Pendekatan ini membuat pemain mampu memahami kompleksitas UHI secara intuitif melalui interaksi langsung dengan representasi spasial, melampaui sekadar angka dan grafik konvensional.



Gambar 3. 12 UI ketika fetching data berhasil



Tahap berikutnya, **Heatmap & Marker Preview State**, menampilkan *heatmap* dari berbagai variabel lingkungan seperti NDVI, NDWI, NDBI, NDMI, dan UDI. Pemain dapat mengganti variabel secara *real-time* tanpa jeda, sebelum memasuki tahap utama **Exploration State** seperti gambar 3.13. Pada tahap ini, pemain menjelajahi lingkungan 3D menggunakan dua mode navigasi: ground mode untuk observasi detail dan fly mode untuk melihat pola panas dari udara. Saat pemain mendekati hotspot, sistem otomatis beralih ke **Monitoring & Analysis State**, yang menampilkan nilai LST serta variabel terkait. Pemain memilih variabel yang dianggap paling berpengaruh, lalu sistem memberikan umpan balik dan memperbarui progres pemantauan.



Gambar 3. 13 UI Monitoring dengan dilengkapi VR Left Hand UI

Antarmuka utama permainan berupa **VR Hand UI Panel** ditempatkan di tangan kiri pemain. Panel ini menampilkan intensitas UHI, jumlah hotspot yang telah dianalisis, sisa waktu, serta pengaturan kondisi waktu (pagi, siang, malam,

hujan). Desain panel dibuat ringkas agar tetap informatif tanpa mengganggu eksplorasi. FSM kemudian mengatur transisi menuju **Completion State**, yaitu tahap akhir ketika waktu permainan berakhir atau seluruh hotspot selesai dimonitor. Pada tahap ini, sistem menampilkan ringkasan hasil permainan meliputi skor, jumlah hotspot yang diamati, dan rata-rata intensitas UHI.

## 5. Evaluasi *Usability* dengan SUS

Evaluasi tahap akhir dilakukan untuk menilai **tingkat kenyamanan dan kemudahan penggunaan sistem**. Metode yang digunakan adalah **SUS**, yaitu metode evaluasi standar yang mengukur persepsi pengguna terhadap sistem berdasarkan sepuluh pernyataan. Evaluasi *usability* merupakan langkah esensial dalam menilai tingkat kegunaan suatu sistem berbasis teknologi, termasuk visualisasi berbasis VR yang dikembangkan dalam penelitian ini. Menurut Nuriman et al. (2020), pengujian dan evaluasi *usability* tidak hanya penting untuk meningkatkan kepuasan pengguna, tetapi juga berfungsi sebagai dasar dalam merancang sistem interaktif yang lebih intuitif, efisien, dan mudah dipahami. Dengan demikian, proses evaluasi *usability* menjadi fondasi utama dalam memahami sejauh mana sistem VR yang dikembangkan mampu memenuhi ekspektasi pengguna, baik dari sisi fungsionalitas maupun pengalaman pengguna secara keseluruhan. Uji *usability* dilakukan terhadap beberapa responden yang mencoba sistem VR secara langsung. Mereka diminta untuk menyelesaikan skenario permainan hingga akhir, kemudian mengisi kuesioner SUS yang mencakup aspek kemudahan penggunaan, konsistensi tampilan, kejelasan

informasi, serta kepuasan keseluruhan. Nilai SUS dihitung menggunakan formula standar:

$$SUS\ Score = \sum (x - 1) \times 2.5 \quad (3.9)$$

Setiap item dalam SUS dinilai menggunakan skala Likert 5 poin dengan skor akhir berkisar antara 0–100. Berdasarkan interpretasi Brooke (1996), skor  $SUS \geq 68$  dianggap menunjukkan sistem dengan tingkat *usability* yang baik pada Tabel 3.3.

Tabel 3. 5 Variabel Skala Likert (Budiaji, 2013)

No	Skala Likert	Nilai
1	Sangat Tidak Setuju (STS)	1
2	Tidak Setuju (TS)	2
3	Netral	3
4	Setuju (S)	4
5	Sangat Setuju (ST)	5

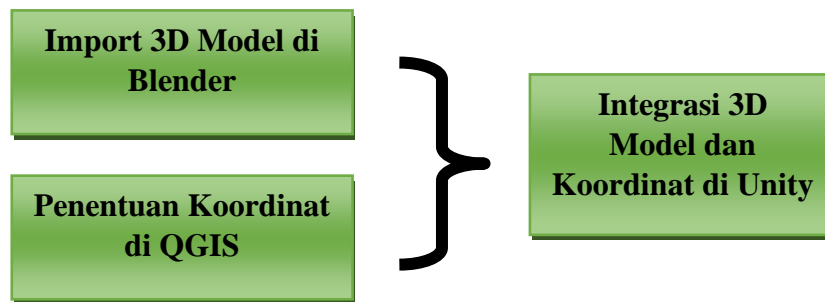
SUS dikenal sebagai metode evaluasi yang sederhana, reliabel, dan telah terbukti efektif untuk memberikan gambaran umum mengenai *usability* suatu sistem (Brooke, 1996). Untuk memperdalam interpretasi hasil, setiap item dalam SUS dapat dikategorikan ke dalam lima aspek utama *usability* yang dikemukakan oleh Nielsen (2012), yaitu: Learnability, Efficiency, Memorability, *Errors*, dan Satisfaction. Pengelompokan ini membantu dalam memetakan persepsi pengguna terhadap dimensi-dimensi utama dari pengalaman penggunaan sistem secara lebih spesifik, sekaligus memberikan masukan yang lebih mendalam bagi proses pengembangan sistem ke depannya. Hasil dari tahap ini adalah sistem visualisasi adaptif yang mampu menampilkan fenomena UHI secara ***real-time***, **interaktif**,

dan **edukatif**. Sistem mengintegrasikan prediksi suhu dari API, menyesuaikannya dengan data lingkungan, lalu merepresentasikannya dalam bentuk *heatmap* tiga dimensi di lingkungan VR. Dengan adanya mekanisme FSM, pemain dapat berinteraksi secara dinamis, menjelajahi area panas kota, dan memahami pengaruh setiap variabel lingkungan terhadap peningkatan suhu permukaan. Sementara itu, hasil evaluasi SUS memberikan ukuran kuantitatif terhadap pengalaman pengguna, memastikan bahwa sistem tidak hanya informatif tetapi juga mudah digunakan dan menyenangkan.

## BAB IV

### PERANCANGAN DAN PENGEMBANGAN 3D CITY MODEL

Perancangan dan pengembangan *3D City Model* merupakan tahapan krusial dalam penelitian ini karena menjadi fondasi utama bagi sistem visualisasi adaptif UHI berbasis *Virtual Reality* (VR). Model tiga dimensi yang dibangun berfungsi sebagai media representasi spasial hasil prediksi *Land Surface Temperature* (LST) yang dihasilkan oleh algoritma *SVR*, sehingga mampu menampilkan kondisi suhu permukaan kota secara visual, interaktif, dan dinamis. Proses pengembangan model dilakukan melalui tiga tahap utama yang saling terintegrasi, yaitu: (1) Pembuatan model tiga dimensi kota menggunakan Blender dengan dukungan plugin BIOSM (Blender–OpenStreetMap), yang berfungsi untuk mengonversi data spasial dua dimensi dari OpenStreetMap menjadi model 3D yang realistis dan efisien, (2) Penentuan titik koordinat dan pemetaan suhu permukaan tanah menggunakan QGIS (Quantum GIS), yang berperan dalam menghubungkan data spasial dunia nyata dengan titik referensi virtual yang digunakan dalam simulasi, serta (3) **Integrasi model spasial dan data suhu ke dalam Unity 3D**, yang membentuk lingkungan VR interaktif dan mampu beradaptasi secara *real-time* terhadap variasi nilai suhu dari hasil prediksi model. Rangkaian proses lengkap pengembangan model dapat dilihat pada **Gambar 4.1**, yang memperlihatkan bagaimana data spasial dari OSM diolah menjadi model visual melalui Blender, diperkaya dengan data koordinat dari QGIS, dan akhirnya diintegrasikan ke Unity sebagai simulasi adaptif tiga dimensi

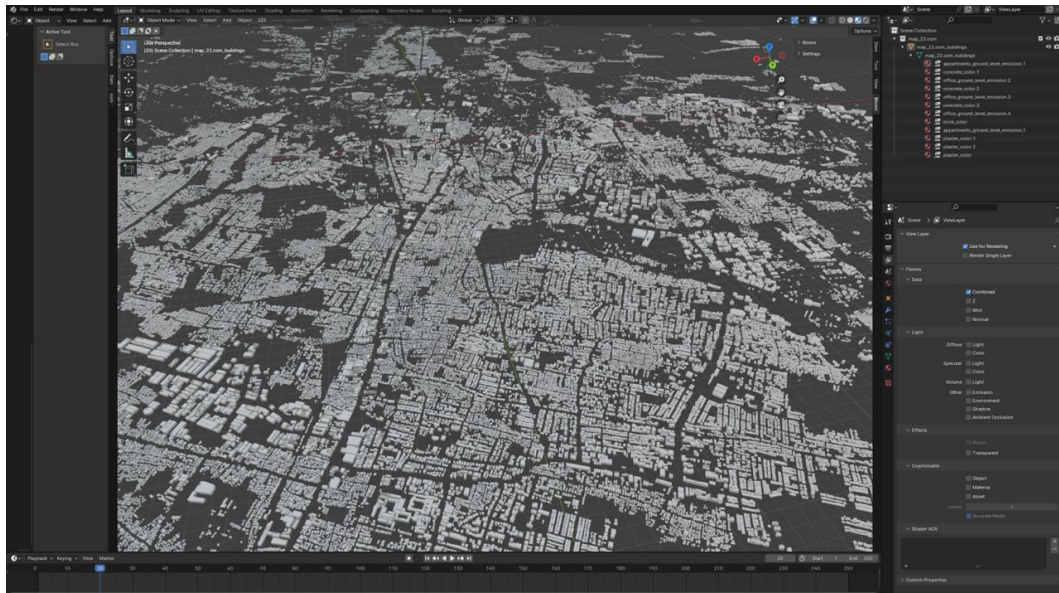


Gambar 4. 1 Tahapan umum perancangan dan pengembangan 3D

#### 4.1 Pembuatan 3D City Model

Tahapan awal dimulai dengan pembuatan model tiga dimensi kota menggunakan perangkat lunak Blender yang dilengkapi dengan plugin BLOSM (Blender–OpenStreetMap). BLOSM memungkinkan pengguna mengimpor data spasial langsung dari OpenStreetMap (OSM) berdasarkan batas koordinat geografis tertentu, sehingga secara otomatis menghasilkan model kota yang sesuai dengan kondisi geografis aslinya. Pemilihan metode ini dilakukan karena Blender bersifat open-source, mendukung pipeline ekspor ke berbagai format kompatibel dengan Unity, serta memungkinkan tingkat kontrol yang tinggi terhadap geometri dan material objek. Dalam penelitian ini, wilayah administratif Kota Malang didefinisikan menggunakan sistem proyeksi WGS 84 (EPSG:4326) agar konsisten dengan sistem koordinat yang digunakan di QGIS dan Unity. Batas minimum dan maksimum koordinat (longitude dan latitude) dimasukkan pada antarmuka BLOSM untuk mengunduh data spasial seperti data bangunan, jalan, vegetasi, dan badan air diunduh secara otomatis. Hasil ekstraksi awal menghasilkan model dasar kota dalam bentuk *polygonal mesh* sederhana dengan struktur geometris menyerupai tata ruang asli. Tampilan hasil ekstraksi awal model 3D Kota Malang

dapat dilihat pada Gambar 4.2, di mana setiap objek bangunan dan jalan berhasil di *generate* secara otomatis dari data OSM.



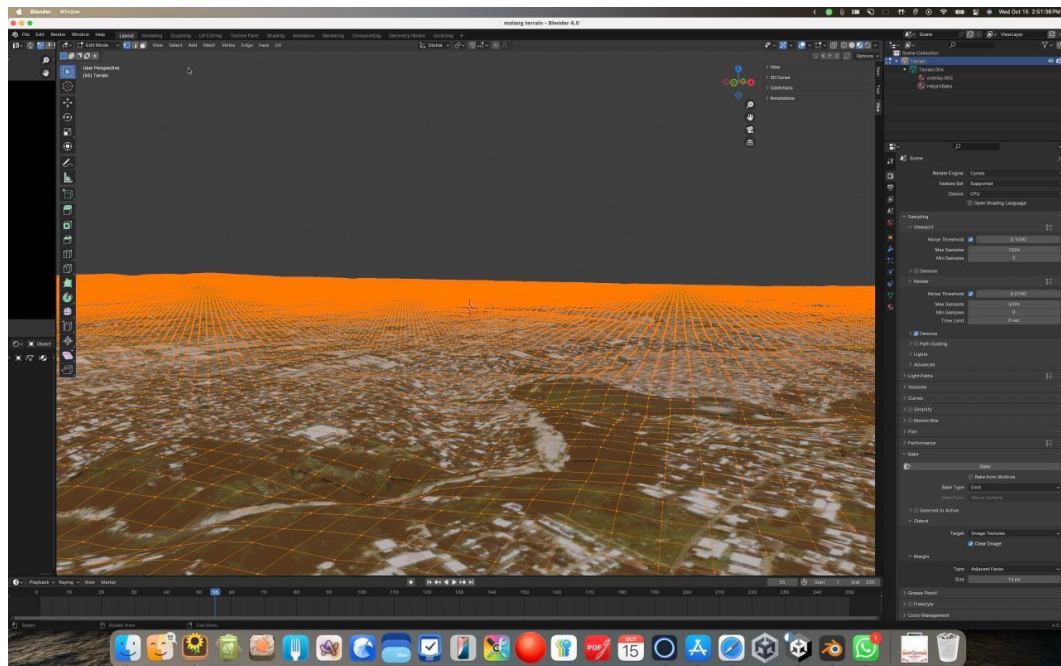
Gambar 4. 2 Tampilan model tiga dimensi Kota Malang di Blender menggunakan plugin BIOSM

Meskipun hasil awal cukup akurat secara spasial, struktur yang dihasilkan masih berupa bidang datar dua dimensi. Oleh karena itu, dilakukan proses penyempurnaan manual pada tahap berikutnya untuk meningkatkan realisme dan performa visualisasi. Langkah-langkah yang dilakukan antara lain:

- Penyesuaian Elevasi Dan Kontur Tanah

Tahapan penyesuaian elevasi dan kontur tanah merupakan langkah penting dalam proses pembentukan terrain yang realistis di Blender. Elevasi suatu wilayah menentukan seberapa akurat simulasi dapat merepresentasikan kondisi geospasial sebenarnya, terutama untuk studi fenomena UHI yang sangat dipengaruhi oleh topografi dan kerapatan permukaan bangunan terhadap radiasi panas. Dalam penelitian ini, data topografi diperoleh dari Digital Elevation Model (DEM)

SRTM (Shuttle Radar Topography Mission) dengan resolusi spasial 30 meter seperti pada gambar 4.3. SRTM dipilih karena memiliki cakupan global, reliabilitas tinggi, dan ketersediaan bebas biaya.



Gambar 4. 3 Generate Elevation terrain menggunakan SRTM

Data DEM SRTM tersebut diunduh dalam format raster GeoTIFF, kemudian diimpor ke QGIS untuk proses preprocessing awal. Selanjutnya, dilakukan normalisasi nilai elevasi dengan metode min-max normalization untuk menghindari perbedaan ketinggian ekstrem yang dapat menyebabkan distorsi saat diimpor ke Blender. Di dalam Blender, raster elevasi yang telah diproses dikonversi menjadi bentuk heightmap, yaitu citra abu-abu (grayscale) di mana setiap piksel mewakili nilai ketinggian. Proses ini memungkinkan Blender untuk mengubah data raster menjadi displacement geometry yang membentuk terrain tiga dimensi. Nilai ketinggian kemudian disesuaikan dengan skala vertikal Kota Malang menggunakan rasio 1 unit Blender = 1 meter agar tetap konsisten dengan sistem koordinat WGS84.



Hasil terrain memperlihatkan karakteristik morfologi Kota Malang yang khas: elevasi rendah di bagian utara seperti kecamatan Blimbing dan Kedungkandang, serta peningkatan ketinggian menuju selatan yaitu wilayah Sukun dan Tidar dengan perbedaan elevasi mencapai  $\pm 400$  meter. Kontur ini penting karena memengaruhi pola sirkulasi udara dan perbedaan suhu permukaan, yang pada akhirnya berdampak pada tingkat intensitas UHI. Untuk menjaga performa rendering, dilakukan proses retopologi mesh pada terrain. Proses ini mengubah mesh hasil displacement yang semula berisi jutaan poligon menjadi struktur yang lebih efisien dengan mempertahankan bentuk dasar kontur tanpa mengorbankan detail visual signifikan.

- **Penerapan material dan tekstur homogen**

Penerapan material dan tekstur homogen merupakan tahapan berikutnya yang berfungsi untuk memperkaya tampilan visual dari model kota, sekaligus meningkatkan realisme lingkungan tanpa menambah beban komputasi yang berlebihan. Dalam konteks penelitian ini, tujuan utama penerapan material bukan hanya untuk memperindah tampilan, tetapi juga untuk memperkuat keterbacaan spasial agar pengguna VR dapat mengenali perbedaan antara jenis permukaan (aspal, vegetasi, bangunan, air) secara intuitif.

Proses teksturisasi dimulai dengan penyusunan atlas tekstur (texture atlas) di Blender. Atlas ini berfungsi untuk menggabungkan beberapa jenis material (misalnya beton, genteng, vegetasi, air) ke dalam satu lembar tekstur tunggal beresolusi tinggi. Pendekatan ini dilakukan untuk menghindari penggunaan

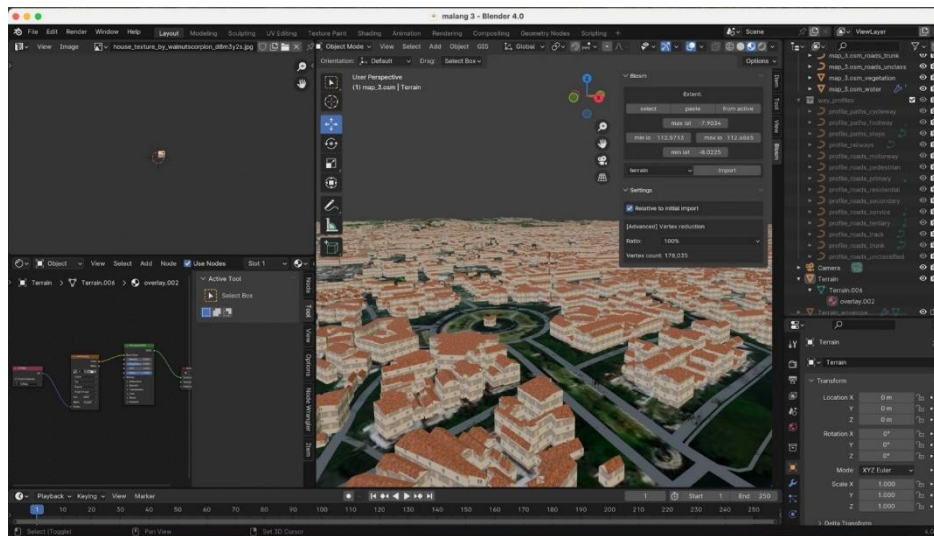
material berbeda pada tiap objek yang dapat meningkatkan jumlah *draw calls* dan memperlambat kinerja sistem di Unity.

Sumber tekstur diperoleh dari dua sumber utama:

1. OpenStreetMap (OSM) untuk peta dasar bangunan dan permukaan jalan;
2. Google Earth (non-komersial, tangkapan citra resolusi tinggi) untuk meniru warna permukaan nyata.

Citra tersebut dikoreksi warnanya terlebih dahulu agar tingkat kecerahan dan saturasi seragam. Kemudian dilakukan proses *UV mapping* di Blender, di mana permukaan setiap objek bangunan diproyeksikan ke atlas tekstur sesuai proporsi bentuknya. Bangunan di area perumahan diberi tekstur genteng berwarna oranye kecokelatan, sementara area pusat kota diberi tekstur atap abu-abu dengan pantulan lebih tinggi untuk meniru sifat reflektif permukaan beton.

Selain itu, material shader diatur menggunakan prinsip *PBR (Physically Based Rendering)* agar respons cahaya lebih realistis di Unity. Setiap material memiliki parameter *albedo* (warna dasar), *roughness* (tingkat kekasaran), dan *metallic* (tingkat pantulan cahaya). Permukaan jalan misalnya memiliki nilai *roughness* tinggi dan *metallic* rendah untuk menampilkan efek pantulan lembut, sedangkan genteng dan kaca memiliki nilai *metallic* tinggi untuk memperkuat reflektansi cahaya matahari virtual.



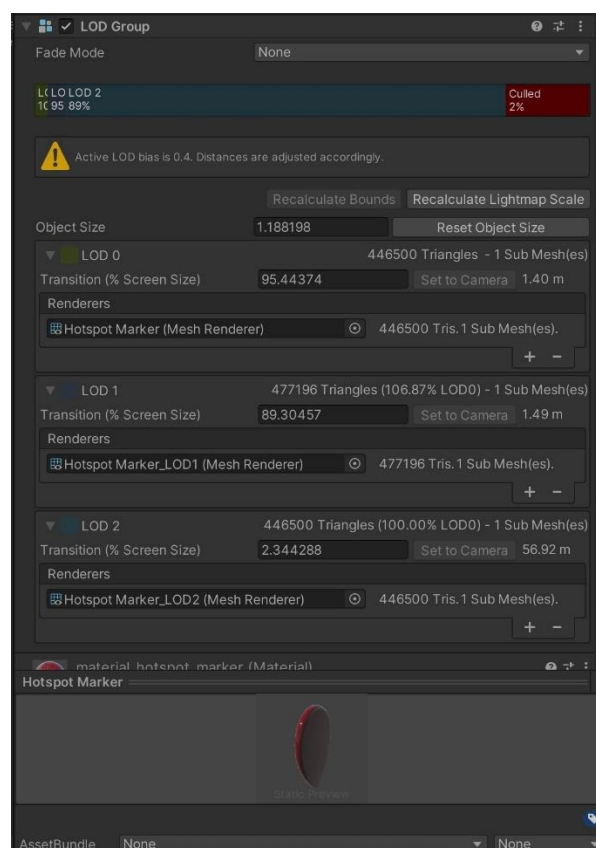
Gambar 4. 4 Penerapan tekstur homogen pada terrain

Gambar 4.4 merupakan tahapan akhir adalah penerapan tekstur homogen pada terrain. Citra satelit resolusi tinggi dari OSM disesuaikan menggunakan proyeksi ortogonal dan kemudian dilapisi ke permukaan terrain. Teknik ini memungkinkan area vegetasi, jalan, dan bangunan tampil dengan perbedaan warna alami tanpa perlu menambahkan model 3D tambahan. Untuk menambah kedalaman visual, diterapkan juga *normal map* dan *ambient occlusion map*, yang membantu menciptakan efek bayangan mikro (*mikro-shading*) pada permukaan tanah dan dinding bangunan. Penerapan material dan tekstur homogen ini menghasilkan tampilan kota yang tidak hanya realistis, tetapi juga efisien. Total penggunaan memori GPU untuk keseluruhan model turun hingga 30% dibandingkan dengan penggunaan material individual, sementara kualitas visual tetap terjaga dengan tingkat detail yang cukup tinggi pada mode VR

- Penerapan *Level of Detail (LOD)* dan *texture baking*

Tahapan ini merupakan salah satu komponen optimasi visual paling penting, terutama untuk memastikan sistem visualisasi tetap berjalan mulus di

lingkungan VR yang memiliki keterbatasan sumber daya komputasi. *Level of Detail (LOD)* adalah teknik yang mengatur tingkat kompleksitas model berdasarkan jarak pandang pengguna terhadap objek. Sementara itu, *texture baking* digunakan untuk menggabungkan efek pencahayaan dan bayangan permanen ke dalam tekstur sehingga mengurangi beban proses rendering waktu nyata (*real-time rendering load*).



Gambar 4. 5 Penerapan LOD

Dalam penelitian ini, penerapan LOD dilakukan dengan membuat tiga versi dari setiap objek utama, yaitu:

- LOD0: model berpoligon tinggi untuk objek yang berada dalam radius < 50 meter dari pengguna VR;

- LOD1: model dengan jumlah poligon 50–60% lebih rendah untuk radius 50–150 meter;
- LOD2: model *low-poly* untuk objek pada jarak >150 meter dari pengguna.

Blender secara otomatis menghasilkan versi LOD1 dan LOD2 menggunakan fitur *decimate modifier*, sementara struktur utama tetap dipertahankan agar bentuk bangunan tidak terdistorsi. Ketika diimpor ke Unity, sistem *LOD Group* digunakan untuk mengatur peralihan antarversi secara otomatis sesuai jarak kamera VR. Teknik ini mampu mengurangi jumlah poligon aktif hingga 70% pada saat pengguna berada di tengah kota, yang secara signifikan menurunkan beban GPU dan menjaga *frame rate* tetap stabil di atas 60 FPS.

Sementara itu, texture baking digunakan untuk menggabungkan hasil pencahayaan global (*global illumination*) dan bayangan statis ke dalam satu peta tekstur (*lightmap*). Proses ini dilakukan di Blender menggunakan *Cycles Renderer* dengan pengaturan *indirect lighting* agar cahaya pantulan antarobjek terekam pada tekstur. Setelah proses baking selesai, hasil *lightmap* disimpan sebagai tekstur tambahan yang menempel pada objek ketika diimpor ke Unity. Keuntungan dari pendekatan ini adalah Unity tidak lagi perlu menghitung pencahayaan kompleks secara *real-time*, karena sebagian besar informasi pencahayaan sudah “dibekukan” dalam tekstur. Hasilnya, sistem dapat mempertahankan visualisasi kota yang terang, realistis, dan memiliki kedalaman bayangan alami tanpa menurunkan kinerja rendering VR.

Kombinasi antara *LOD* dan *texture baking* menjadikan visualisasi 3D Kota Malang tidak hanya realistis, tetapi juga efisien secara teknis. Saat diuji pada perangkat *Meta Quest 2* dan mode desktop, model dapat diputar, ditelusuri, dan diperbesar tanpa penurunan signifikan pada *frame rate*. Dari hasil pengukuran performa, rata-rata *draw calls* berkurang dari 3.200 menjadi hanya sekitar 1.150 per frame setelah optimasi diterapkan, sedangkan penggunaan memori GPU menurun sekitar 40%.

Setelah model selesai disempurnakan, dilakukan validasi spasial dengan membandingkan posisi objek utama seperti bangunan dan jalan terhadap citra Google Maps 3D dan data vektor QGIS. Pengukuran menggunakan jarak Euclidean menunjukkan deviasi rata-rata di bawah 5 meter, yang masih dalam batas toleransi untuk skala representasi kota. Selain itu, validasi visual kualitatif dilakukan dengan membandingkan bentuk dan warna permukaan dengan citra satelit untuk memastikan kesesuaian visual terutama di kawasan padat seperti Klojen dan Blimbing.

Gabungan dari ketiga tahapan di atas, menghasilkan representasi visual Kota Malang yang realistis, efisien, dan mendukung pengalaman imersif di lingkungan VR. Terrain yang berbukit memperlihatkan konteks topografi yang relevan terhadap fenomena panas perkotaan, tekstur homogen menghadirkan realisme spasial tanpa membebani GPU, dan sistem LOD memastikan visualisasi tetap halus serta stabil. Setelah proses pemodelan selesai, model diekspor dalam format .FBX (Filmbox) karena format ini mampu mempertahankan informasi material, hierarki objek, dan sistem koordinat secara utuh, sekaligus memiliki

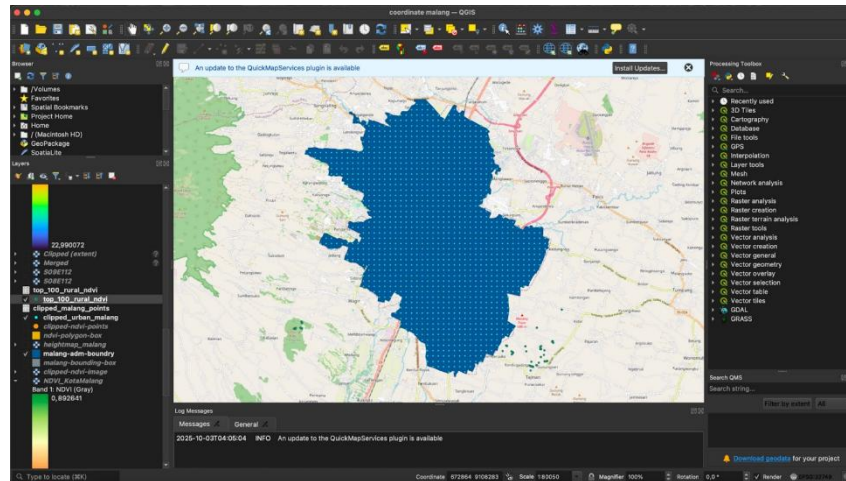
kompatibilitas tinggi dengan Unity. Format ini juga memudahkan integrasi ke Unity, di mana setiap elemen kota seperti bangunan, jalan, dan vegetasi dapat dikontrol secara individual dalam sistem visualisasi.

#### **4.2 Penentuan Titik Koordinat**

Langkah berikut dalam proses perancangan dan pengembangan sistem visualisasi adalah penentuan titik koordinat serta pemetaan data LST ke dalam objek 3D di lingkungan virtual Kota Malang. Proses ini merupakan tahap krusial yang menjembatani antara data geospasial dunia nyata dengan representasi visual di dalam Unity, sekaligus memastikan bahwa setiap nilai hasil prediksi dari model SVR dapat ditransformasikan secara akurat ke elemen-elemen 3D yang ada dalam simulasi.

Tahap awal dilakukan melalui penentuan titik-titik koordinat sampel menggunakan perangkat lunak QGIS. QGIS dipilih karena kemampuannya dalam manajemen data geospasial, analisis spasial, serta fleksibilitas dalam mengintegrasikan berbagai format data, termasuk shapefile (*.shp*, *.shx*, *.dbf*, dan *.prj*). Data batas administratif Kota Malang diimpor menggunakan layer *ADMINISTRASIDESAR\_25K*, yang mencakup lima kecamatan utama: Klojen, Blimbing, Lowokwaru, Sukun, dan Kedungkandang. Setelah batas administratif dimuat, dilakukan pembuatan titik-titik sampel di dalam area tersebut dengan memanfaatkan plugin “Regular Points”. Plugin ini menghasilkan distribusi titik koordinat secara acak atau merata di dalam poligon batas wilayah kota, sehingga seluruh area studi dapat terwakili secara spasial. Distribusi titik yang dihasilkan

dapat dilihat pada Gambar 4.3, yang menunjukkan pemerataan titik koordinat di seluruh wilayah administratif Kota Malang.



Gambar 4. 6 Distribusi titik koordinat representatif Kota Malang menggunakan plugin *Regular Points* di QGIS.

Gambar 4.6 merupakan proses pemerataan titik koordinat yang menghasilkan lebih dari 10.000 titik koordinat awal yang mencakup keseluruhan wilayah Kota Malang. Namun, untuk menjaga efisiensi pemrosesan data dan kinerja sistem pada tahap simulasi di Unity, dilakukan reduksi dan optimasi jumlah titik menjadi sekitar 1.332 titik representatif. Proses optimasi mempertimbangkan pemerataan spasial agar setiap kecamatan memiliki kepadatan titik yang seimbang dan tidak menimbulkan bias visual pada hasil peta termal. Setelah dipastikan bahwa distribusi titik sudah optimal, seluruh data koordinat diekspor ke format CSV (Comma-Separated Values) yang berisi pasangan nilai *latitude* dan *longitude*. File CSV inilah yang menjadi dasar untuk proses *fetching* data suhu secara *real-time* melalui API Open-Meteo, sekaligus menjadi referensi posisi bagi sistem Unity untuk menempatkan nilai suhu pada lokasi yang sesuai.



Tahapan selanjutnya adalah pemetaan data LST hasil prediksi SVR ke objek-objek 3D di dalam model kota. Proses ini dilakukan melalui dua pendekatan utama, yaitu *object-based mapping* dan *grid-based mapping*. Pada *object-based mapping*, setiap bangunan 3D hasil konversi dari data OpenStreetMap (OSM) diidentifikasi berdasarkan posisi spasialnya, kemudian diberikan nilai suhu rata-rata dari piksel raster yang menutupi area bangunan tersebut. Pendekatan ini menghasilkan visualisasi yang sangat detail dan akurat, terutama untuk mode *desktop visualization* yang berfokus pada kejelasan tampilan. Namun, karena jumlah objek 3D di Kota Malang mencapai lebih dari 80.000 unit, metode ini cukup berat untuk dijalankan dalam mode VR yang membutuhkan performa *real-time rendering*. Oleh karena itu, diterapkan *grid-based mapping* sebagai alternatif yang lebih efisien. Dalam pendekatan ini, setiap bangunan pada model 3D hasil konversi dari OpenStreetMap diidentifikasi berdasarkan koordinat pusat (centroid) bangunan. Sistem kemudian mencari nilai suhu rata-rata dari piksel raster (hasil prediksi SVR) yang menutupi area bangunan tersebut. Nilai suhu itu disimpan sebagai atribut objek dan digunakan untuk mengubah warna bangunan secara dinamis melalui shader di Unity. Kelebihan metode ini adalah akurasi tinggi dan detail spasial yang presisi, karena setiap bangunan memiliki nilai suhu tersendiri. Hal ini ideal untuk mode *desktop visualization* atau analisis ilmiah, di mana pengguna memerlukan detail per objek. Namun, metode ini memiliki kelemahan dari sisi performa. Jumlah bangunan di Kota Malang yang mencapai lebih dari 80.000 unit menyebabkan rendering load dan draw calls meningkat tajam, terutama saat dijalankan dalam mode VR.

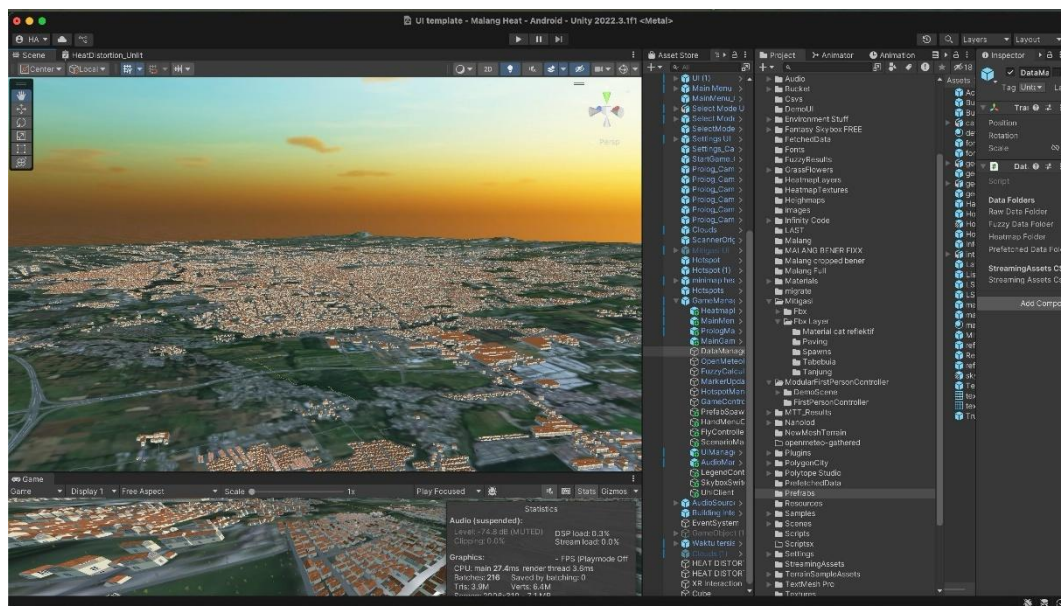
Untuk mode VR, diterapkan Grid-based Mapping karena jauh lebih efisien secara komputasi. Dalam metode ini, wilayah Kota Malang dibagi menjadi grid berukuran 300×300 meter, sesuai dengan resolusi titik sampling dari tahap Regular Points. Setiap grid menerima nilai suhu rata-rata hasil prediksi SVR berdasarkan titik-titik yang berada di dalamnya. Proses penentuan nilai grid dilakukan menggunakan metode interpolasi spasial dengan algoritma Inverse Distance Weighting (IDW). Algoritma ini mengasumsikan bahwa nilai suhu pada titik yang lebih dekat memiliki pengaruh lebih besar daripada titik yang lebih jauh.

Hasil dari interpolasi ini menghasilkan distribusi suhu yang halus (*smooth gradient*), sehingga perbedaan suhu antargrid terlihat gradual tanpa terkesan terfragmentasi. Hasil akhir dari pemetaan suhu divisualisasikan sebagai *heatmap* tiga dimensi di atas permukaan kota. Warna pada setiap area menyesuaikan nilai suhu hasil prediksi SVR, sehingga pengguna dapat mengamati pola termal di seluruh wilayah Kota Malang dengan jelas.

#### **4.3 Integrasi Model dan Data ke Unity**

Tahapan terakhir adalah integrasi model 3D dan data suhu ke dalam Unity. Proses ini mencakup penggabungan model spasial (.FBX) dan data CSV titik koordinat hasil QGIS. Model .FBX diimpor menggunakan 3D Asset Importer dan dikonversi menjadi Unity Terrain menggunakan plugin Mesh to Terrain. Tekstur citra satelit resolusi tinggi digunakan untuk melapisi terrain, sementara setiap bangunan dan elemen kota diberi standard shader material agar kompatibel dengan sistem adaptive shader yang akan digunakan untuk memvisualisasikan

variasi suhu permukaan tanah secara dinamis. Tampilan hasil integrasi dapat dilihat pada Gambar 4.7, yang menunjukkan lingkungan virtual Kota Malang dengan skala spasial dan detail visual yang mendekati kondisi nyata. Sistem ini juga mengimplementasikan pencahayaan adaptif dan warna termal (thermal *overlay*) yang akan berubah mengikuti data suhu yang diperoleh dari server.



Gambar 4. 7 Integrasi model 3D Kota Malang ke dalam Unity 3D untuk sistem visualisasi UHI berbasis VR.

Selanjutnya, dilakukan pemrograman sistem komunikasi antara Unity dan server menggunakan C# untuk mengakses API prediksi suhu berbasis FastAPI. Proses pengambilan data suhu dari server dijalankan melalui HTTP POST request sebagaimana dijelaskan dalam pseudocode pertama. Dalam lampiran 2 pseudocode Unity Web Request ke Server, sistem menentukan endpoint API (/gee/predict-from-csv), menyiapkan request body berformat JSON berisi tanggal prediksi, window days, backfill days, serta urutan fitur model (feature order). Data kemudian dikirim ke server dengan timeout tertentu, dan hasil response berupa file CSV diterima serta disimpan secara otomatis untuk kemudian dipetakan pada

terrain Unity. Pseudocode ini juga memuat mekanisme validasi hasil, seperti pengecekan apakah response kosong atau apakah file CSV berhasil disimpan.

Setelah Unity menerima data CSV hasil prediksi, tahap selanjutnya dijalankan di sisi server sesuai dengan lampiran 2. Pada pseudocode Data Input, Validasi, dan Persiapan Data GEE, server membaca file CSV yang dikirim oleh Unity, melakukan validasi keberadaan kolom wajib (longitude, latitude, type), serta memastikan jumlah titik tidak melebihi 2500 agar proses prediksi tetap efisien. Seluruh titik diubah menjadi FeatureCollection di Google Earth Engine (GEE), kemudian dibuat region of interest (ROI) dengan buffer 2 km untuk memastikan cakupan area analisis yang memadai. Sistem menentukan rentang waktu analisis berdasarkan tanggal yang diminta, serta mempersiapkan data citra yang akan diambil dari Earth Engine.

Tahap berikutnya dijelaskan dalam pseudocode Pemrosesan GEE, Prediksi Model, dan Output, di mana sistem melakukan komposit citra satelit dalam dua jendela waktu: narrow window dan wide window. Celah data yang hilang diisi menggunakan metode unmasking antara kedua citra tersebut. Data suhu permukaan (Land Surface Temperature) kemudian diekstraksi dan digabung dengan feature stack (NDVI, NDBI, NDWI, dsb.) untuk membentuk set data lengkap. Titik-titik koordinat sampel digunakan untuk melakukan sampling nilai setiap fitur, dan hasilnya dikonversi menjadi DataFrame sebelum diprediksi menggunakan model SVR. Hasil prediksi kemudian dinormalisasi kembali menjadi satuan derajat Celsius dan disimpan dalam format CSV untuk dikirim kembali ke Unity.

Ketiga pseudocode tersebut membentuk siklus data yang berjalan secara berkelanjutan dan adaptif: Unity mengirim permintaan prediksi berdasarkan koordinat dari QGIS, server memproses data melalui GEE dan SVR, kemudian hasil prediksi suhu dikembalikan untuk divisualisasikan secara *real-time* di lingkungan virtual Kota Malang. Dengan mekanisme ini, sistem dapat memperbarui visualisasi suhu secara dinamis setiap kali nilai baru diambil dari API, menjadikan 3D City Model tidak sekadar representasi statis, melainkan simulasi adaptif yang merespons perubahan lingkungan secara langsung.

## **BAB V**

### ***DATA REMOTE SENSING DAN MODEL SVR***

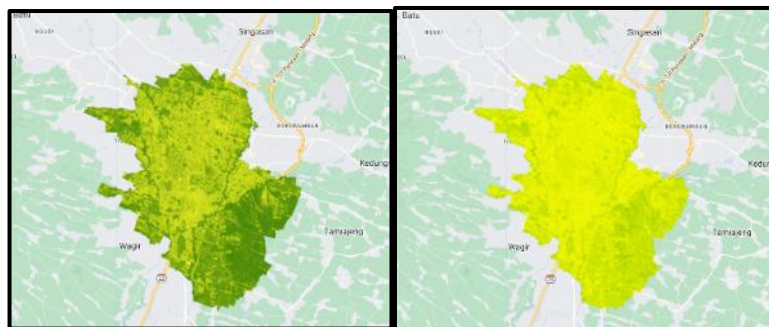
#### **5.1 Pengumpulan Data dan Preprocessing**

Tahapan awal dalam penelitian ini dimulai dengan proses pengumpulan data penginderaan jauh (remote sensing) yang bersumber dari platform Google Earth Engine (GEE). Data diambil dari tiga satelit utama, yaitu Sentinel-2 Level-2A, Landsat-8 OLI/TIRS Collection 2 Level 1, serta NASA SRTM Digital Elevation Model (DEM) dengan resolusi spasial 30 meter. Area kajian penelitian difokuskan pada wilayah administratif Kota Malang, Jawa Timur. Setiap data citra satelit yang diperoleh tidak langsung digunakan dalam proses analisis. Sebelum dimanfaatkan lebih lanjut, data tersebut harus melalui tahapan pra-pengolahan (preprocessing) untuk memastikan kualitas dan akurasi datanya. Dua langkah penting yang dilakukan pada tahap ini adalah Cloud Masking (penyaringan awan) dan Feature Scaling (penskalaan fitur). Kedua proses ini bertujuan untuk menghilangkan gangguan dari awan serta menyesuaikan skala nilai antar variabel agar dapat diolah secara optimal pada tahap pemodelan berikutnya.

##### **5.1.1 Cloud Masking**

Proses Cloud Masking merupakan langkah esensial dalam pengolahan citra satelit, bertujuan untuk membersihkan data dari pengaruh awan dan bayangan awan yang dapat mengaburkan informasi sebenarnya dari permukaan bumi. Piksel-piksel yang tertutup awan biasanya memiliki nilai reflektansi yang tidak konsisten, sehingga dapat mengganggu hasil analisis jika tidak disaring terlebih dahulu. Pada penelitian ini, proses Cloud Masking dilakukan terhadap citra

Sentinel-2 Level 2A menggunakan algoritma berbasis Scene Classification Layer (SCL) yang mampu mengidentifikasi dan menghapus piksel awan serta bayangan awan. Hasil perbandingan antara citra sebelum dan sesudah Cloud Masking ditampilkan pada Gambar 5.1, yang menunjukkan citra vegetasi wilayah Kota



Gambar 5. 1 Perbandingan visualisasi citra vegetasi menggunakan Sentinel-2 sebelum dan setelah dilakukan Cloud Masking

Malang.

Gambar 5.1 menunjukkan hasil perbandingan nilai piksel vegetasi sebelum dan setelah penerapan Cloud Masking pada citra Sentinel-2 Level 2. Dari hasil perbandingan tersebut terlihat bahwa citra setelah dilakukan Cloud Masking memiliki tampilan visual yang lebih bersih dan akurat. Warna hijau yang mendominasi menunjukkan area dengan tingkat vegetasi tinggi, sedangkan warna kekuningan di area pusat kota menunjukkan rendahnya nilai vegetasi akibat dominasi permukaan terbangun seperti bangunan dan jalan. Sebaliknya, citra sebelum dilakukan Cloud Masking memperlihatkan warna yang cenderung kuning menyeluruh, menandakan adanya pengaruh awan dan bayangan awan yang terekam oleh sensor satelit. Dengan demikian, proses Cloud Masking terbukti efektif dalam meningkatkan kualitas citra sebelum dilakukan analisis lanjutan.

### 5.1.2 Feature Scalling

Langkah berikutnya dalam tahap pra-pengolahan data adalah Feature Scalling. Tahap ini dilakukan untuk menyeragamkan skala nilai antar variabel yang berasal dari berbagai kanal (band) citra satelit. Proses ini penting agar setiap variabel memiliki kontribusi yang seimbang dalam model pembelajaran mesin (machine learning). Dalam penelitian ini, Feature Scalling dilakukan pada beberapa band dari citra Sentinel-2, yaitu Band 3 (Green), Band 4 (Red), Band 8 (Near-Infrared/NIR), Band 11 (Short-Wave Infrared 1), dan Band 12 (Short-Wave Infrared 2). Dari kombinasi band-band tersebut kemudian dihitung sejumlah indeks spektral, antara lain NDVI (Normalized Difference Vegetation Index), NDBI (Normalized Difference Built-up Index), NDMI (Normalized Difference Moisture Index), NDWI (Normalized Difference Water Index), serta UDI (*Urban Density Index*). Tabel 5.1 menampilkan contoh hasil perhitungan sebelum dan sesudah dilakukan Feature Scalling. Melalui tahap ini, setiap fitur yang berbeda skala nilainya berhasil disesuaikan sehingga siap digunakan untuk proses analisis dan pemodelan dengan metode SVR.

*Tabel 5.1 Feature Calculation*

Sebelum Feature Calculation						Setelah Feature Calculation				
Data	Band 3 (G)	Band 4 (R)	Band 8 (N)	Band 11 (S1)	Band 12 (S2)	NDVI	NDBI	NDMI	NDWI	UDI
1	0.0741	0.0668	0.2385	0.1902	0.1289	0.0741	0.0668	0.23855	0.19025	0.1289
2	0.0785	0.0737	0.2559	0.1981	0.1418	0.5528	-0.2868	0.24862	-0.5305	0.2413
3	0.1146	0.1173	0.1987	0.2211	0.1979	0.2574	-0.0020	0.15357	-0.2685	1

## 5.2 Normalisasi Data



Setelah proses Feature Scaling, langkah selanjutnya adalah normalisasi data untuk menyamakan rentang nilai antar variabel input agar tidak terjadi dominasi oleh salah satu variabel. Normalisasi dilakukan menggunakan metode Z-Score Standardization, sebagaimana dinyatakan dalam Persamaan (3.1). Metode ini mengonversi setiap nilai variabel menjadi bentuk standar dengan rata-rata nol dan deviasi standar satu. Dengan demikian, seluruh variabel seperti NDVI, NDMI, NDBI, NDWI, UDI, Elevation, dan LST (Land Surface Temperature) memiliki kontribusi yang seimbang dalam model. Tabel 5.2 merupakan contoh hasil data sebelum dan sesudah normalisasi. Hasil tabel tersebut menunjukkan bahwa setelah dilakukan normalisasi, seluruh variabel telah berada dalam rentang distribusi yang seragam, sehingga siap digunakan untuk proses pemodelan regresi.

Tabel 5. 2 Sebelum dan Sesudah Normalisasi data

Sebelum Normalisasi						
NDVI	NDMI	NDBI	NDWI	UDI	Elevation	LST
-0.015	-0.160	0.210	0.016	0.931	491	38.1
-0.018	-0.215	0.272	0.004	0.966	466	39.6
-0.006	-0.162	0.183	0.005	0.931	423	40.2
-0.011	-0.028	0.065	0.024	0.931	431	41.7
-0.015	-0.226	0.248	-0.007	0.966	463	42.9
Sesudah Normalisasi						
NDVI	NDMI	NDBI	NDWI	UDI	Elevation	LST
-1.671	-1.056	1.298	1.956	0.885	0.226	0.377
-1.682	-1.335	1.497	1.903	0.972	-0.272	0.754
-1.639	-1.067	1.212	1.908	0.885	-0.109	0.933
-1.657	-0.390	0.834	1.991	0.885	-0.970	1.315
-1.671	-1.390	1.420	1.855	0.972	-0.332	1.648

### 5.3 Data Splitting

Langkah berikutnya adalah pemisahan dataset (data splitting) menjadi dua bagian utama, yaitu data training dan data testing. Data training digunakan untuk melatih model SVR agar mampu mengenali pola hubungan antar variabel,

sedangkan data testing berfungsi untuk menguji kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Pemisahan data dilakukan secara acak dengan menggunakan metode random state untuk menjaga konsistensi hasil setiap kali proses dijalankan. Dalam penelitian ini, dilakukan tiga skenario pembagian rasio data yang berbeda untuk melihat pengaruh variasi rasio terhadap performa model, yaitu 70:30, 80:20, dan 90:10. Hasil pemisahan ini selanjutnya digunakan dalam proses tuning parameter dan pemilihan kernel terbaik untuk model SVR. Tabel 5.3 menunjukkan perbedaan rasio splitting datanya:

*Tabel 5. 3 Rasio Splitting Data*

Skenario	Rasio	$\Sigma$ Data Training	$\Sigma$ Data Test
Skenario 1	70:30	5950	2550
Skenario 2	80:20	6800	1700
Skenario 3	90:10	7650	850

Selain skenario pembagian dataset, terdapat juga sub-skenario tuning parameter. Tuning parameter digunakan untuk mencari nilai parameter yang optimal sehingga SVR memberikan keseimbangan yang baik antara kesalahan klasifikasi dan margin maksimum sesuai dengan karakteristik data yang ada. Nantinya, setiap skenario pada Tabel 5.3 akan diuji dengan sub-skenario yang ada pada Tabel 5.4.

*Tabel 5. 4 Skenario Fungsi Kernel*

Sub-Skenario	Fungsi Kernel	Parameter <i>Hyperplane</i>	Nilai Parameter
1	Kernel Linear	$C$ ( <i>Cost</i> )	$10^{-1}, 1, 10^1$
		$\varepsilon$ ( <i>Epsilon</i> )	$10^{-3}, 10^{-2}, 10^{-1}$
2	Kernel <i>Polynomial</i>	$C$ ( <i>Cost</i> )	$10^{-1}, 1, 10^1$
		$\varepsilon$ ( <i>Epsilon</i> )	$10^{-3}, 10^{-2}, 10^{-1}$
		$d$ ( <i>Degree</i> )	2, 4, 8
3	Kernel Radial Basis Function	$C$ ( <i>Cost</i> )	$10^{-1}, 1, 10^1$
		$\varepsilon$ ( <i>Epsilon</i> )	$10^{-3}, 10^{-2}, 10^{-1}$
		$\gamma$ ( <i>Gamma</i> )	$10^{-1}, 1, 10^1$

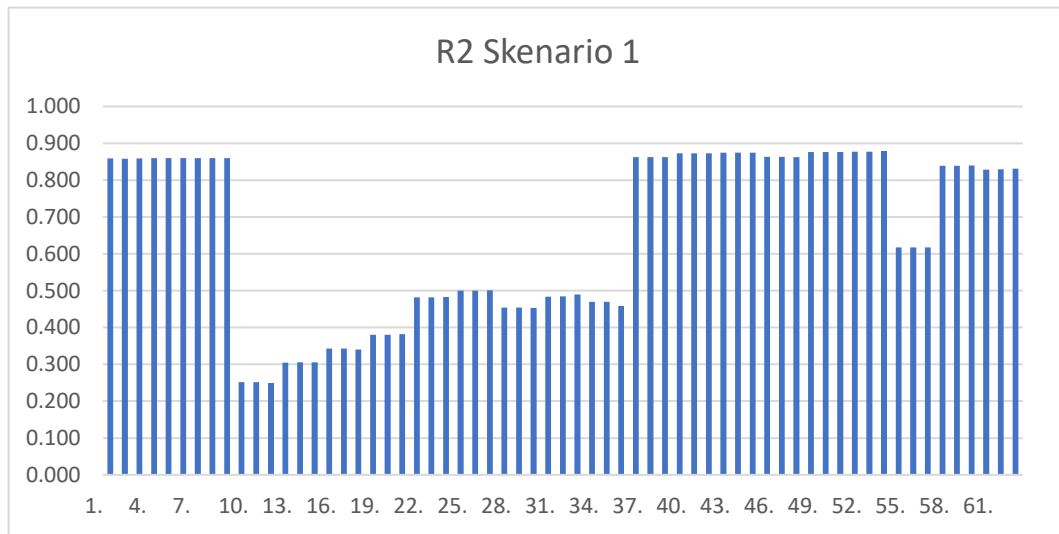
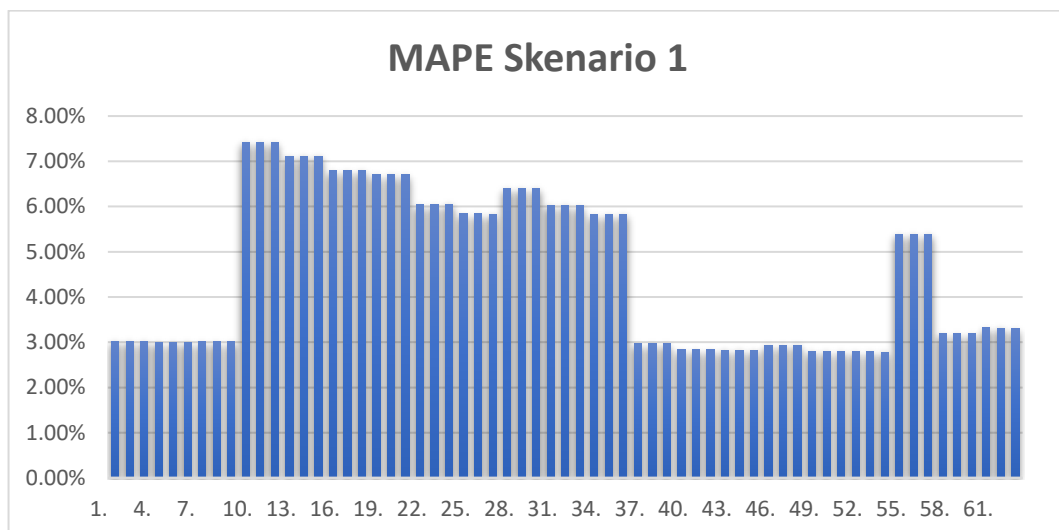
## 5.4 Hasil Pengujian Model SVR

Tahap akhir dari proses ini adalah evaluasi performa model SVR yang digunakan untuk memprediksi LST berdasarkan variabel-variabel hasil penginderaan jauh. Evaluasi dilakukan dengan membandingkan nilai aktual ( $y_{test}$ ) dengan nilai prediksi ( $y_{pred}$ ) yang dihasilkan oleh model. Model diuji dengan berbagai kombinasi kernel dan parameter hasil hyperparameter tuning untuk memperoleh performa terbaik. Ukuran yang digunakan untuk menilai kinerja model diutamakan pada MAPE, karena metrik ini mampu menunjukkan tingkat kesalahan prediksi dalam bentuk persentase, sehingga mudah diinterpretasikan. Klasifikasi nilai MAPE mengacu pada Tabel 2.2, di mana semakin kecil nilai MAPE, semakin baik tingkat akurasi model dalam melakukan prediksi. Dengan pendekatan ini, performa model SVR dapat dievaluasi secara objektif dan komprehensif untuk mengetahui sejauh mana model mampu menggambarkan pola hubungan antara variabel-variabel spasial terhadap fenomena UHI di Kota Malang.

### 5.4.1 Hasil pengujian Skenario 1

Pada skenario pertama, pembagian data dilakukan dengan komposisi 70% sebagai data latih (training data) dan 30% sebagai data uji (testing data). Dari total keseluruhan dataset, diperoleh 5.950 data latih dan 2.550 data uji. Proporsi ini dipilih untuk menguji sejauh mana model SVR mampu melakukan generalisasi terhadap data baru ketika diberikan jumlah data latih yang relatif moderat. Dengan kata lain, skenario ini dirancang untuk mengukur kemampuan model dalam mengenali pola hubungan antar variabel input dan memprediksi LST pada kondisi

di mana proporsi data uji lebih besar dibandingkan skenario lain. Evaluasi kinerja model dilakukan dengan menggunakan dua metrik utama, yaitu  $R^2$  (Coefficient of Determination) dan MAPE (Mean Absolute Percentage *Error*). Ketiga metrik tersebut digunakan secara komplementer untuk menggambarkan tingkat akurasi prediksi serta konsistensi model dalam merepresentasikan hubungan antara variabel masukan dan keluaran. Gambar 5.3, dan Gambar 5.4 berikut menampilkan hasil evaluasi model SVR pada Skenario 1 untuk setiap kombinasi kernel dan parameter.

Gambar 5.2 R<sup>2</sup> Skenario 1

Gambar 5.3 MAPE Skenario 1

Berdasarkan hasil visualisasi pada Gambar 5.2, terlihat bahwa nilai  $R^2$  mengalami fluktuasi yang cukup tajam pada berbagai konfigurasi kernel dan parameter. Rentang nilai  $R^2$  menunjukkan perbedaan signifikan antara kernel yang digunakan, menandakan sensitivitas model terhadap pemilihan fungsi kernel dan parameter regulasi. Secara umum, nilai  $R^2$  tertinggi diperoleh pada pengujian dengan kernel RBF (Radial Basis Function), khususnya pada rentang pengujian ke-37 hingga ke-65, di mana model mencapai nilai  $R^2$  maksimum sebesar 0.878.

Hal ini menunjukkan bahwa model dengan kernel RBF mampu menjelaskan sekitar 87,8% variasi data aktual dalam prediksi nilai LST.

Sementara itu, hasil pada Gambar 5.3 memperlihatkan variasi nilai MAPE, yang merepresentasikan rata-rata kesalahan absolut dalam satuan persentase. Model dengan kernel RBF menunjukkan performa paling unggul dengan nilai MAPE sebesar 2.78%, menandakan bahwa rata-rata kesalahan prediksi terhadap data aktual sangat kecil, hanya sekitar 2,78%. Berdasarkan klasifikasi akurasi prediksi menurut Lewis (1982), nilai MAPE di bawah 5% tergolong *highly accurate forecast*, sehingga dapat disimpulkan bahwa model SVR pada konfigurasi ini memiliki tingkat akurasi yang sangat tinggi.

Sebaliknya, hasil pengujian dengan kernel Polynomial menunjukkan performa yang jauh lebih rendah dibandingkan kernel lainnya. Nilai  $R^2$  hanya mencapai 0.469 dengan MAPE sebesar 5.82%, menandakan bahwa model gagal menyesuaikan pola hubungan antar variabel secara optimal. Hal ini terutama disebabkan oleh penggunaan parameter *degree* yang terlalu tinggi ( $d = 8$ ), sehingga model mengalami *overfitting*, yaitu terlalu menyesuaikan diri pada data latih dan kehilangan kemampuan generalisasi pada data uji. Fenomena ini juga tercermin pada kenaikan nilai MSE yang signifikan pada kernel Polynomial, yang menunjukkan bahwa tingkat kesalahan prediksi meningkat secara drastis.

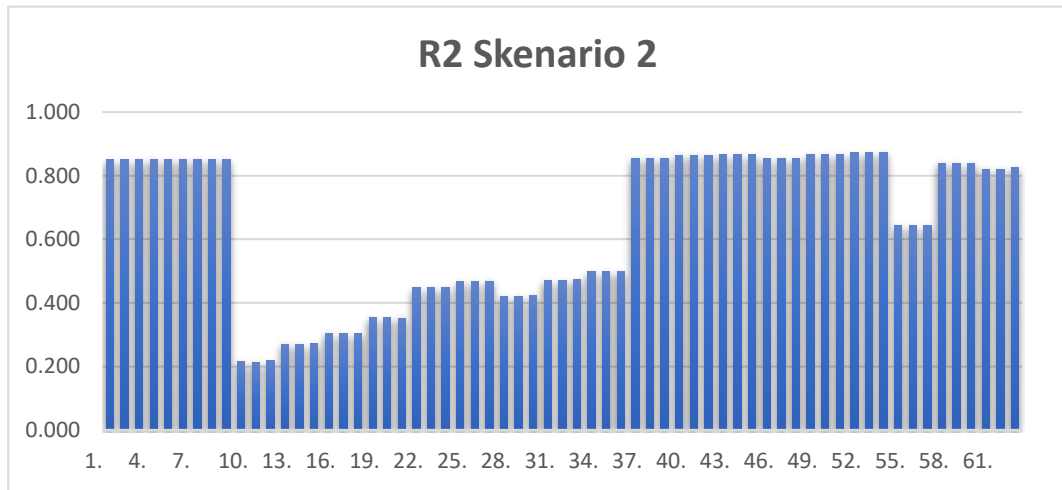
Sementara itu, kernel Linear memberikan hasil yang cukup stabil dengan  $R^2$  sebesar 0.860 dan MAPE sebesar 3.00%. Meskipun akurasinya sedikit di bawah RBF, kernel ini tetap menunjukkan performa yang baik dan konsisten, khususnya dalam kasus hubungan antar variabel yang relatif linear. Namun, keterbatasannya

terletak pada ketidakmampuannya menangani hubungan non-linear yang kompleks antara variabel-variabel citra satelit seperti NDVI, NDMI, NDBI, dan elevasi, yang berperan besar dalam menentukan variasi suhu permukaan tanah.

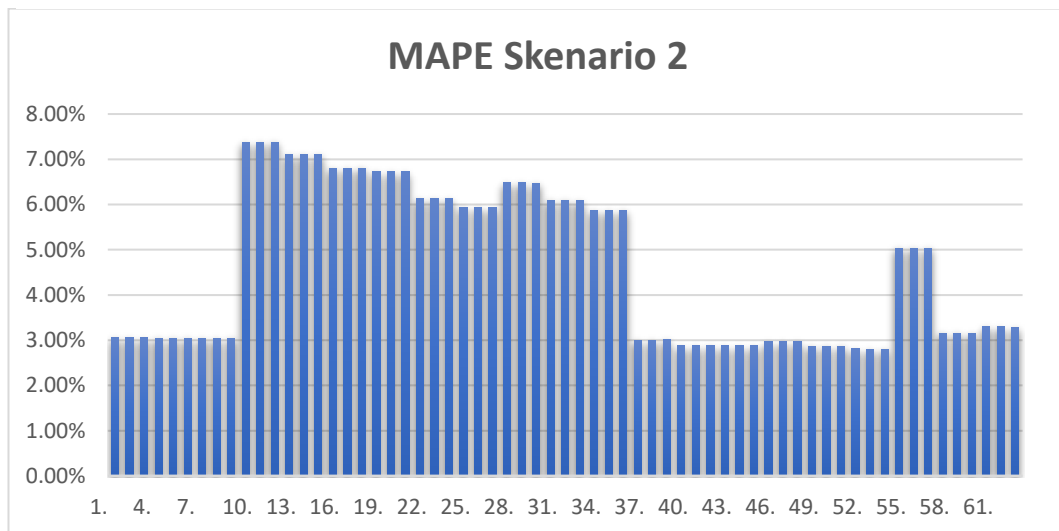
Berdasarkan hasil ini, dapat disimpulkan bahwa pada rasio 70:30, kernel RBF menghasilkan performa optimal dengan konfigurasi  $C = 10$ ,  $\varepsilon = 0.1$ , dan  $\gamma = 1$ , menghasilkan nilai  $R^2$  sebesar 0.878 dan MAPE sebesar 2.78%. Kinerja ini menunjukkan bahwa pendekatan SVR dengan kernel RBF mampu menangkap karakteristik kompleks dari fenomena UHI melalui hubungan non-linear antar variabel, sekaligus memberikan hasil prediksi yang akurat dan stabil.

#### **5.4.2 Hasil pengujian Skenario 2**

Pada skenario kedua, data latih ditingkatkan menjadi 80% (6.800 data), dan data uji berkurang menjadi 20% (1.700 data). Tujuan skenario ini adalah untuk mengamati efek peningkatan jumlah data latih terhadap akurasi model dan kestabilan hasil prediksi. Gambar 5.4 dan Gambar 5.5 berikut menunjukkan perbandingan performa model SVR pada skenario kedua berdasarkan tiga metrik evaluasi utama, yaitu  $R^2$ , dan MAPE. Selanjutnya dilakukan komparasi antara NN model B1 dan B2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 5.4 hingga Gambar 5.8.



Gambar 5. 4  $R^2$ -Skenario 2



Gambar 5. 5 MAPE Skenario 3

Dari hasil yang ditampilkan pada Gambar 5.4, terlihat bahwa nilai  $R^2$  mengalami fluktuasi pada berbagai konfigurasi kernel dan parameter, namun secara umum menunjukkan tren yang konsisten dengan skenario pertama. Nilai  $R^2$  tertinggi kembali diperoleh pada penggunaan kernel RBF (Radial Basis Function), terutama pada pengujian ke-37 hingga ke-50, di mana model menunjukkan  $R^2$  mencapai sekitar 0.873. Nilai ini menandakan bahwa model mampu menjelaskan 87,3% variasi data aktual terhadap hasil prediksi, sehingga menunjukkan



kemampuan model dalam memetakan hubungan non-linear antar variabel dengan sangat baik.

Pada bagian awal pengujian (sekitar iterasi ke-1 hingga ke-7), model menunjukkan performa tinggi dengan nilai  $R^2$  di atas 0.8, menandakan kestabilan model pada konfigurasi parameter tertentu. Namun, performa menurun drastis pada iterasi ke-10 hingga ke-20, di mana nilai  $R^2$  turun ke kisaran 0.2–0.4. Hal ini mengindikasikan bahwa kombinasi parameter tertentu, seperti nilai  $C$  dan  $\varepsilon$  yang tidak optimal, membuat model gagal menggeneralisasi hubungan antar variabel. Seiring dengan proses *tuning* parameter, performa model kembali meningkat dan stabil pada pengujian dengan kernel RBF, menunjukkan bahwa kernel ini mampu beradaptasi terhadap data kompleks dengan baik.

Selanjutnya, hasil MAPE (Mean Absolute Percentage Error) yang ditampilkan pada Gambar 4.8 memberikan ukuran tingkat kesalahan relatif dalam bentuk persentase, yang memudahkan interpretasi performa model. Nilai MAPE terbaik diperoleh pada kernel RBF dengan kisaran 2.80%, yang menunjukkan bahwa rata-rata kesalahan prediksi terhadap data aktual hanya sebesar 2,8%. Berdasarkan klasifikasi akurasi prediksi menurut Lewis (1982), nilai MAPE di bawah 5% tergolong kategori “sangat akurat” (*highly accurate prediction*). Hal ini menunjukkan bahwa model SVR dengan kernel RBF memiliki kemampuan prediksi yang sangat tinggi terhadap nilai suhu permukaan tanah di wilayah penelitian.

Sebaliknya, kernel Linear menunjukkan performa sedikit menurun dibandingkan pada skenario pertama, dengan nilai  $R^2$  sebesar 0.851 dan MAPE

sebesar 3.04%. Penurunan kecil ini menunjukkan bahwa peningkatan kompleksitas data latih tidak serta merta meningkatkan performa kernel Linear, karena kernel ini bekerja optimal pada hubungan antar variabel yang bersifat linier. Dengan meningkatnya jumlah data non-linear, kernel Linear cenderung kesulitan menyesuaikan pola variasi suhu permukaan yang dipengaruhi oleh vegetasi, kerapatan bangunan, serta elevasi.

Sementara itu, kernel Polynomial masih menunjukkan performa terendah, dengan MAPE sebesar 5.87% dan  $R^2$  hanya mencapai 0.467. Nilai tersebut memperkuat temuan dari skenario pertama bahwa kernel Polynomial kurang cocok digunakan untuk kasus dengan tingkat non-linearitas tinggi seperti hubungan antara indeks spektral dan suhu permukaan. Model Polynomial juga menunjukkan MSE yang sangat tinggi di atas 8.000, menandakan ketidakstabilan dalam prediksi akibat kompleksitas fungsi Polynomial yang berlebihan.

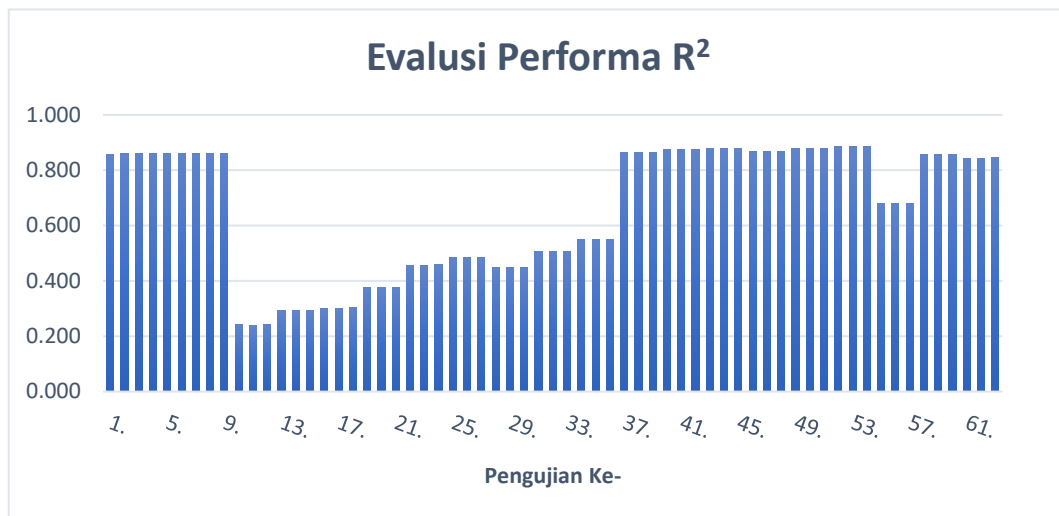
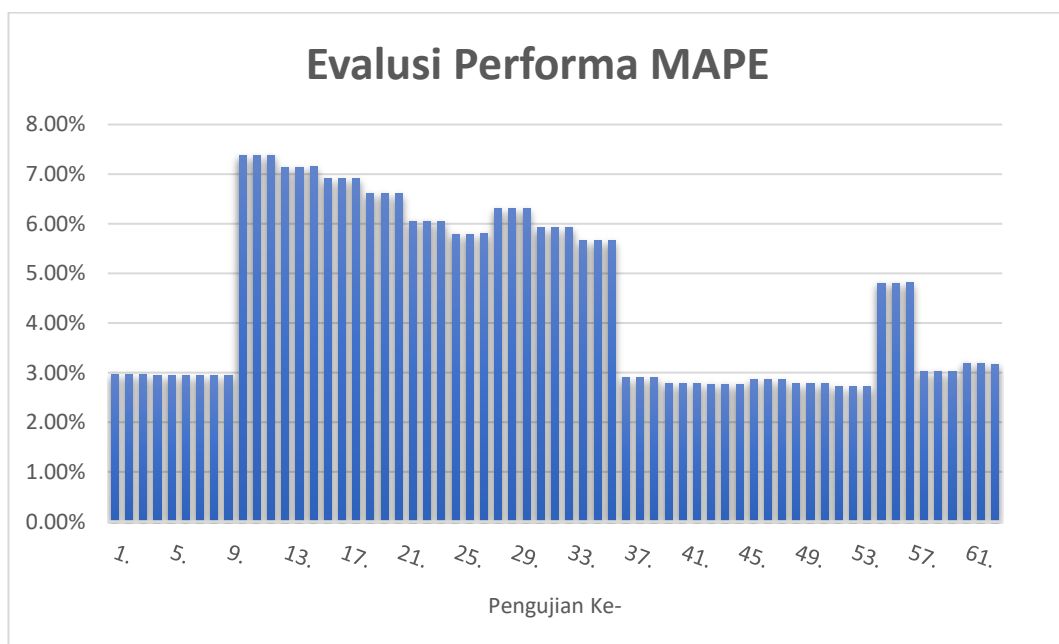
Jika dibandingkan dengan hasil Skenario 1, performa model pada Skenario 2 tidak menunjukkan peningkatan signifikan dari sisi akurasi, namun menunjukkan peningkatan dalam kestabilan model. Hal ini terlihat dari sebaran nilai  $R^2$ , MSE, dan MAPE yang cenderung lebih seragam dan stabil dibandingkan sebelumnya. Peningkatan jumlah data latih sebesar 10% membuat model lebih mengenali variasi data secara lebih menyeluruh tanpa mengorbankan kemampuan generalisasi

Dengan demikian, dapat disimpulkan bahwa peningkatan proporsi data latih dari 70% menjadi 80% memberikan dampak positif terhadap kestabilan model, meskipun tidak secara signifikan meningkatkan akurasi keseluruhan.

Kernel RBF dengan konfigurasi hyperparameter  $C = 10$ ,  $\varepsilon = 0.1$ , dan  $\gamma = 1$  kembali terbukti sebagai kombinasi paling optimal, dengan hasil akhir berupa  $R^2$  sebesar 0.873 dan MAPE sebesar 2.80%. Hasil ini menegaskan bahwa kernel RBF memiliki kemampuan yang konsisten dalam memetakan hubungan non-linear antar variabel hasil penginderaan jauh terhadap nilai suhu permukaan tanah, sekaligus menunjukkan kemampuan generalisasi yang kuat terhadap data uji baru.

### 5.4.3 Hasil pengujian Skenario 3

Pada skenario ketiga, pembagian data dilakukan dengan komposisi 90% sebagai data latih (training data) dan 10% sebagai data uji (testing data). Pembagian ini menghasilkan total 7.650 data latih dan 850 data uji. Skenario ini dirancang untuk mengamati performa model ketika diberikan hampir seluruh dataset sebagai data pelatihan, dengan data uji dalam jumlah minimal. Gambar 5.6 dan Gambar 5.7 menampilkan hasil evaluasi performa model berdasarkan masing-masing metrik tersebut.

Gambar 5. 6  $R^2$  Skenario 3

Gambar 5. 7 MAPE Skenario 3

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 4.9, model dengan kernel RBF (Radial Basis Function) kembali menunjukkan performa paling dominan dan stabil dibandingkan kernel lainnya. Nilai  $R^2$  tertinggi mencapai 0.887, yang berarti model mampu menjelaskan sekitar 88,7% variasi

data aktual terhadap hasil prediksi. Nilai  $R^2$  ini mendekati angka 0.9, sehingga dapat dikatakan bahwa model memiliki kemampuan prediksi yang sangat baik dalam memetakan hubungan non-linear antara variabel-variabel input hasil penginderaan jauh terhadap suhu permukaan tanah.

Dari grafik tersebut terlihat bahwa performa model di awal pengujian (sekitar pengujian ke-1 hingga ke-7) sudah menunjukkan nilai  $R^2$  tinggi di atas 0.8, menandakan bahwa model berhasil menggeneralisasi pola data dengan baik sejak tahap awal. Sementara pada pengujian ke-10 hingga ke-25, performa sempat menurun dengan nilai  $R^2$  turun ke kisaran 0.3–0.5. Penurunan ini mengindikasikan adanya kombinasi parameter kernel yang kurang optimal, terutama pada kernel Polynomial dengan degree tinggi yang cenderung *overfitting* terhadap data latih. Setelah dilakukan pengujian lebih lanjut dengan kernel RBF, performa meningkat kembali secara signifikan dan bertahan stabil hingga akhir pengujian (pengujian ke-37–ke-61), dengan  $R^2$  konsisten di kisaran 0.85–0.89.

Nilai MAPE (Mean Absolute Percentage Error) pada Gambar 4.11 memperkuat hasil evaluasi dua metrik sebelumnya. Kernel RBF menghasilkan MAPE sebesar 2.71%, menandakan bahwa rata-rata kesalahan prediksi hanya sekitar 2,71% dari nilai aktual. Nilai ini termasuk dalam kategori “sangat akurat” (highly accurate prediction) menurut klasifikasi Lewis (1982). Performa kernel Linear berada sedikit di bawahnya dengan MAPE sebesar 2.95%, yang masih tergolong sangat baik. Sementara kernel Polynomial menunjukkan MAPE sebesar 5.66%, memperlihatkan kecenderungan kesalahan prediksi yang lebih besar dan konsisten dengan hasil pada dua skenario sebelumnya. Secara keseluruhan, hasil

pengujian menunjukkan bahwa kernel RBF kembali menjadi pilihan terbaik dengan kombinasi parameter  $C = 10$ ,  $\varepsilon = 0.1$ , dan  $\gamma = 1$ . Nilai  $R^2$  sebesar 0.887, MSE sebesar 1.625, dan MAPE sebesar 2.71% menunjukkan bahwa model memiliki keseimbangan optimal antara akurasi dan kestabilan prediksi.

Jika dibandingkan antar skenario, peningkatan proporsi data latih dari 70% (Skenario 1) menjadi 90% (Skenario 3) tidak menghasilkan lonjakan besar dalam akurasi, namun memberikan peningkatan konsistensi dan kestabilan hasil prediksi. Nilai  $R^2$  meningkat secara bertahap dari  $0.878 \rightarrow 0.873 \rightarrow 0.887$ . Dengan demikian, dapat disimpulkan bahwa model SVR dengan kernel RBF telah mencapai titik konvergensi performa pada proporsi data latih sebesar 90%, di mana peningkatan data latih selanjutnya kemungkinan tidak akan memberikan peningkatan signifikan dalam akurasi, tetapi akan tetap memperkuat kestabilan hasil prediksi. Temuan ini membuktikan bahwa kernel RBF merupakan kernel paling efektif dan efisien untuk pemodelan suhu permukaan tanah berbasis data penginderaan jauh, karena kemampuannya dalam mempelajari hubungan non-linear yang kompleks secara konsisten pada berbagai proporsi data latih dan uji.

## **5.5 Analisis Perbandingan Antar Skenario**

Setelah dilakukan serangkaian pengujian menggunakan tiga skenario pembagian data 70:30, 80:20, dan 90:10 dapat disimpulkan bahwa performa model SVR menunjukkan kestabilan dan konsistensi yang sangat baik di seluruh konfigurasi, terutama pada kernel RBF. Secara umum, peningkatan proporsi data latih tidak menghasilkan lonjakan besar dalam akurasi, namun memberikan

dampak positif terhadap peningkatan kestabilan model dan penurunan tingkat kesalahan prediksi secara berkelanjutan.

Tabel 5. 5 Perbandingan hasil pengujian terbaik pada setiap skenario

Skenario	Kernel Terbaik	R <sup>2</sup>	MSE	MAPE
1	RBF	0.878	1.772	2.78%
2	RBF	0.873	1.797	2.80%
3	RBF	0.887	1.625	2.71%

Tabel 5.4 merupakan tabel perbandingan hasil pengujian terbaik antar skenario, kernel RBF secara konsisten menunjukkan hasil terbaik dengan nilai R<sup>2</sup> yang tinggi dan MAPE yang rendah. Pada Skenario 1 (70:30), kernel RBF mencapai R<sup>2</sup> = 0.87 dan MAPE = 2.78%. Nilai ini mengalami sedikit peningkatan pada Skenario 2 (80:20) dengan R<sup>2</sup> = 0.873 dan MAPE = 2.80%, serta kembali meningkat secara signifikan pada Skenario 3 (90:10) dengan R<sup>2</sup> = 0.887 dan MAPE = 2.71%.

Tren ini memperlihatkan bahwa semakin besar jumlah data latih yang digunakan, model semakin stabil dan kesalahan prediksi semakin kecil. Hal ini menunjukkan bahwa model SVR berbasis kernel RBF mampu mencapai konvergensi performa pada proporsi data latih sekitar 80–90%, di mana peningkatan jumlah data latih tidak lagi meningkatkan akurasi secara signifikan, tetapi memperkuat keandalan hasil prediksi. Sebaliknya, kernel Linear menunjukkan performa yang stabil tetapi tidak sebaik RBF. Nilai R<sup>2</sup> berkisar antara 0.851 hingga 0.860 dengan MAPE sekitar 2.95–3.04%. Kernel ini memiliki

keunggulan pada kestabilan hasil dan kecepatan komputasi, namun kurang mampu memetakan pola non-linear yang kompleks seperti hubungan antara vegetasi, kelembapan tanah, kerapatan bangunan, dan elevasi terhadap suhu permukaan tanah. Oleh karena itu, meskipun hasilnya cukup baik, kernel Linear tidak mampu menandingi fleksibilitas kernel RBF dalam mempelajari distribusi data non-linear yang dominan pada data penginderaan jauh.

Sementara itu, kernel Polynomial kembali menunjukkan performa terendah di seluruh skenario, dengan  $R^2$  hanya berkisar antara 0.46–0.55 dan MAPE antara 5.6–5.9%. Hal ini disebabkan oleh kompleksitas fungsi Polynomial yang tinggi, terutama pada nilai degree besar, yang menyebabkan *overfitting* terhadap data latih dan penurunan kemampuan generalisasi pada data uji. Nilai MSE yang jauh lebih tinggi dibanding kernel lain memperkuat indikasi bahwa Polynomial kernel tidak cocok digunakan untuk model prediksi suhu permukaan berbasis data citra satelit yang bersifat heterogen dan non-linear. Hasil pengujian ketiga skenario menunjukkan bahwa Kernel RBF merupakan kernel paling optimal, dengan kinerja terbaik pada seluruh konfigurasi pembagian data. Peningkatan jumlah data latih dari 70% hingga 90% tidak secara signifikan meningkatkan akurasi, namun meningkatkan kestabilan model dan menurunkan nilai MSE secara konsisten. Kernel Linear cocok digunakan pada kondisi hubungan antar variabel yang bersifat mendekati linear, sedangkan kernel Polynomial sebaiknya dihindari untuk data spasial dengan karakteristik non-linear yang kuat.



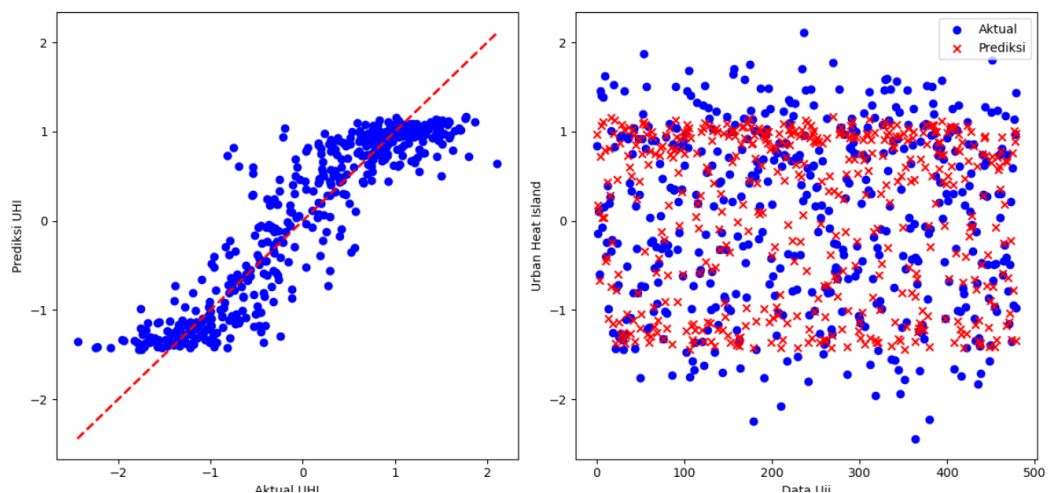
Dengan demikian, hasil keseluruhan membuktikan bahwa model SVR dengan kernel RBF dan konfigurasi parameter  $C = 10$ ,  $\varepsilon = 0.1$ , dan  $\gamma = 1$  merupakan konfigurasi paling efisien dan akurat untuk memprediksi LST di wilayah perkotaan. Kombinasi ini tidak hanya menghasilkan nilai  $R^2$  tertinggi (0.887 pada Skenario 3) dan MAPE terendah (2.71%), tetapi juga menunjukkan kestabilan performa yang kuat pada setiap variasi rasio pembagian data. Hasil ini menegaskan bahwa SVR-RBF mampu menangkap hubungan non-linear kompleks antara variabel-variabel hasil penginderaan jauh, seperti NDVI, NDBI, NDWI, NDMI, dan Elevation, terhadap suhu permukaan tanah. Model ini juga menunjukkan potensi besar untuk diterapkan dalam studi lanjutan terkait fenomena UHI, khususnya dalam konteks pemodelan dan visualisasi spatio-temporal menggunakan sistem berbasis data satelit resolusi tinggi.

## 5.6 Hasil Prediksi Model SVR

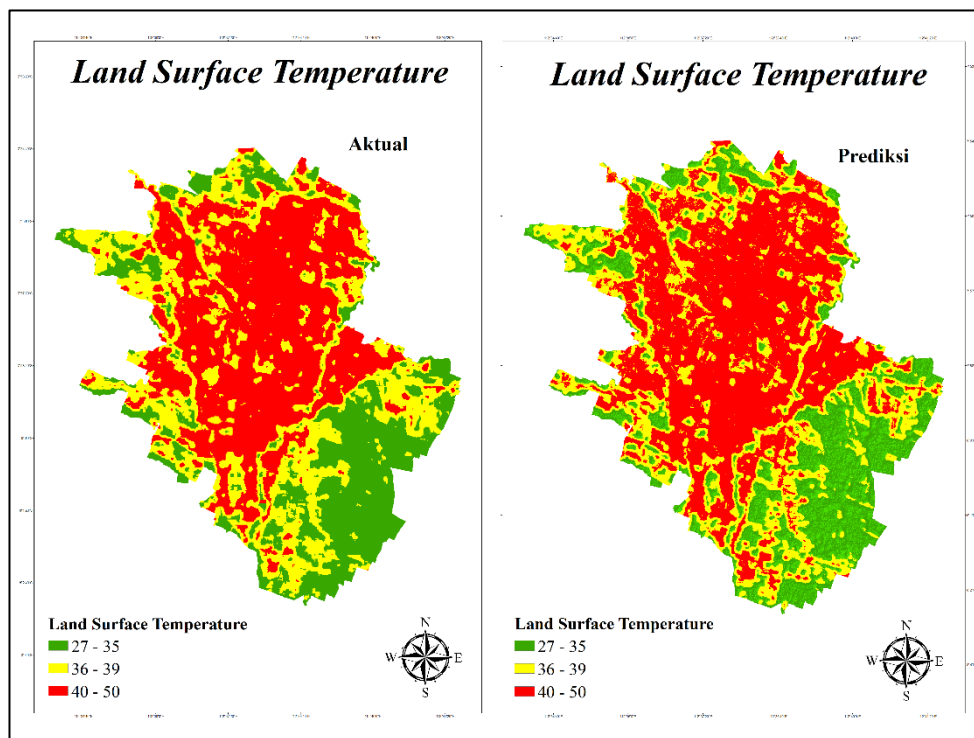
Setelah mengetahui rasio, kernel dan hyperparameter yang terbaik serta bersifat konsisten yaitu menggunakan rasio 90:10 dengan menggunakan kernel linier dan nilai hyperparameter  $C = 10$ ,  $E = 0.1$ . kemudian dilakukan denormalisasi untuk mengembalikan nilai hasil. Proses ini akan mengubah nilai hasil prediksi yang telah dinormalisasi menggunakan *Z-Score* ke dalam skala nilai yang sesungguhnya (suhu dalam °Celcius) sehingga dapat dibandingkan secara langsung dengan data aktual yang sudah ada sebelumnya. Berikut ini perbandingan data aktual dan hasil prediksi yang ditunjukkan pada Tabel 5.5, Scatter plot hasil prediksi pada Gambar 5.8, beserta perbandingan visualisasi citranya yang ditunjukkan pada Gambar 5.9.

Tabel 5.5 Perbandingan Data Aktual dan Hasil Prediksi

No	LST (°C)	
	Data Aktual	Data Prediksi
1	35.4	38.2
2	41.6	37.9
3	33.1	36.1
...	...	...
1454	31.8	30.7
1455	34.1	38.0



Gambar 5. 8 Scatter Plot Hasil Prediksi



Gambar 5.9 Perbandingan citra LST aktual dan hasil prediksi

Perbedaan yang terlihat antara citra aktual LST dan hasil prediksi dapat dijelaskan bahwa citra hasil prediksi menunjukkan luasnya daerah yang masuk dalam kategori suhu tinggi (ditunjukkan dengan warna merah) lebih banyak daripada yang sebenarnya diamati dalam citra aktual. Meskipun demikian, rentang suhu yang diprediksi tidak jauh berbeda, menunjukkan bahwa meskipun model cenderung memperkirakan suhu yang lebih tinggi secara relatif di berbagai daerah, dibuktikan adanya perbedaan antara suhu tertinggi dan terendah yang diprediksi tidak begitu besar. Perbedaan ini dikarenakan adanya karakteristik data yang digunakan dalam pelatihan model, berupa variabel-variabel yang digunakan menunjukkan keberagaman kondisi lingkungan, serta penerapan metode SVR dan penggunaan kernel dan parameter digunakan dalam proses prediksi.

## **BAB VI**

### **ANALISIS EVALUASI PERFORMA MODEL**

Bab ini membahas analisis mendalam terhadap performa model SVR terbaik yang diperoleh dari serangkaian eksperimen pada bab sebelumnya. Berdasarkan hasil pengujian komparatif terhadap tiga jenis kernel Linear, Polynomial, dan RBF model dengan kernel RBF terbukti memberikan hasil paling optimal dalam memprediksi fenomena UHI di wilayah Kota Malang. Model ini menunjukkan nilai koefisien determinasi ( $R^2$ ) rata-rata sebesar 0.879, Mean Absolute Percentage *Error* (MAPE) sebesar 2.76%, dan Mean Square *Error* (MSE) sebesar 1.731, yang menggambarkan kombinasi ideal antara akurasi tinggi, kesalahan rendah, dan stabilitas prediksi antar-skenario.

Keunggulan model RBF bukan hanya tercermin pada metrik numerik, tetapi juga dari kemampuan adaptifnya dalam menangkap hubungan non-linear antara variabel lingkungan yang mempengaruhi suhu permukaan tanah (LST), seperti Normalized Difference Vegetation Index (NDVI), Normalized Difference Built-up Index (NDBI), Normalized Difference Moisture Index (NDMI), *Urban Density Index* (UDI), dan elevasi. Kompleksitas hubungan antarvariabel ini membuat kernel linier kurang mampu merepresentasikan pola aktual, sedangkan RBF mampu membangun pemisahan multidimensi yang lebih fleksibel dan lokal

#### **6.1 Analisis Pemilihan Kernel RBF sebagai Model Terbaik**

##### **6.2.1 Perbandingan Kinerja Antar Kernel**

Berdasarkan hasil pengujian yang diuraikan pada Bab V, kernel RBF *consistently outperform* kernel lainnya dengan menghasilkan nilai MAPE dan MSE yang paling rendah pada seluruh skenario pembagian data. Secara umum, kernel RBF mencatat MAPE rata-rata 2.76% dengan deviasi standar 0.00065, jauh lebih rendah dibanding kernel Linear (3.01%) dan Polynomial (6.42%). Hasil ini mengindikasikan bahwa kernel RBF memiliki kemampuan yang lebih baik dalam mengaproksimasi hubungan kompleks antar fitur lingkungan. Persamaan ini memproyeksikan data dari ruang input berdimensi rendah ke ruang fitur berdimensi sangat tinggi, di mana hubungan non-linear antar variabel dapat diwakilkan oleh garis lurus (hyperplane). Parameter  $\gamma$  (gamma) mengontrol tingkat kelengkungan ruang tersebut. Mekanisme ini memungkinkan RBF untuk mengenali pola non-linear yang tersembunyi dalam hubungan kompleks antar faktor UHI seperti vegetasi, kepadatan bangunan, kelembapan, dan elevasi.

Sebaliknya, kernel Linear hanya membentuk hyperplane tunggal yang memisahkan data berdasarkan fungsi linier, sehingga tidak mampu menangkap variasi lokal yang signifikan. Kernel Polynomial memang bersifat non-linear, namun sangat sensitif terhadap nilai degree, yang menyebabkan risiko *overfitting* bila terlalu tinggi dan *underfitting* bila terlalu rendah. Pada penelitian ini, kernel Polynomial menunjukkan fluktuasi tinggi pada hasil MAPE, menandakan kurang stabilnya model terhadap perubahan parameter.

Selain itu, kernel RBF juga unggul dalam menangani fenomena multi-collinearity antar variabel sebuah karakteristik umum pada data penginderaan jauh, di mana fitur-fitur seperti NDVI dan NDMI sering memiliki korelasi yang

tinggi. Kernel RBF mampu mengelola kondisi ini karena mekanisme pemetaan ke ruang fitur berdimensi tinggi mengurangi efek kolinearitas, berbeda dengan kernel Linear yang rentan terhadap hal tersebut. Dengan mempertimbangkan baik aspek teoritis maupun empiris, dapat disimpulkan bahwa kernel RBF paling tepat digunakan pada kasus pemodelan suhu permukaan berbasis citra satelit yang kompleks secara spasial dan spektral seperti penelitian ini.

### 6.2.2 Stabilitas dan Kemampuan Generalisasi Model

Kualitas model prediksi tidak hanya diukur dari akurasi, tetapi juga dari stabilitas dan kemampuan generalisasi terhadap variasi data. Dalam penelitian ini, tiga skenario pembagian data digunakan untuk mengevaluasi konsistensi model. Hasilnya menunjukkan bahwa fluktuasi MAPE antar-skenario sangat kecil (selisih maksimum 0.07%), dengan nilai  $R^2$  meningkat seiring bertambahnya data latih ( $0.878 \rightarrow 0.887$ ) dan MSE yang menurun ( $1.772 \rightarrow 1.625$ ).

Kestabilan ini mengindikasikan bahwa model mampu mempelajari pola umum (general patterns) tanpa bergantung pada contoh data tertentu. Model tidak menunjukkan indikasi *overfitting*, meskipun proporsi data latih mencapai 90%. Hal ini memperlihatkan kemampuan RBF dalam menemukan struktur dasar data tanpa menghafal *noise* atau *outlier*. Kemampuan generalisasi ini penting karena data citra satelit dan data meteorologi memiliki sifat temporal yang dinamis. Perubahan musiman, kondisi atmosfer, atau variasi waktu pengambilan citra dapat menyebabkan pergeseran distribusi data (data drift). Model RBF yang stabil terhadap variasi semacam ini berarti berpotensi diterapkan untuk prediksi *real-*

*time* atau aplikasi adaptif, misalnya integrasi ke sistem pemantauan UHI di platform VR dengan data dari API Open-Meteo.

## **6.2 Analisis Pengaruh Hyperparameter terhadap Performa Model**

### **6.2.1 Parameter Cost (C)**

Parameter Cost (C) berfungsi mengatur keseimbangan antara kesalahan pelatihan dan kompleksitas model. Nilai C yang besar menyebabkan penalti tinggi untuk kesalahan, menghasilkan model dengan margin sempit dan sensitivitas tinggi terhadap data pelatihan. Sebaliknya, nilai C yang kecil memberi margin lebar, memungkinkan model mengabaikan beberapa kesalahan kecil demi generalisasi lebih baik. Pada penelitian ini, nilai  $C = 10$  memberikan performa terbaik untuk semua kernel, terutama RBF. Ketika nilai C diturunkan menjadi 1 dan 0.1, MAPE meningkat 0.7–0.8%. Ini menunjukkan bahwa model dengan margin yang lebih sempit (C tinggi) mampu membedakan variasi spektral yang halus antar area permukaan tanah dan vegetasi, yang penting dalam konteks UHI. Namun demikian, C terlalu tinggi dapat menyebabkan *overfitting*; stabilitas performa pada  $C = 10$  menunjukkan bahwa distribusi data latih seimbang dan *noise* relatif rendah, sehingga model tetap mampu menjaga generalisasi. Selain itu, dalam konteks spasial, nilai C besar membantu menangani edge case seperti area transisi antara vegetasi dan permukiman yang seringkali sulit diklasifikasikan. Sensitivitas ini memperkuat kemampuan model untuk menangkap gradien termal yang halus di batas-batas zona UHI.

### **6.2.2 Parameter Epsilon ( $\epsilon$ )**

Parameter Epsilon ( $\epsilon$ ) mendefinisikan lebar zona toleransi kesalahan ( $\epsilon$ -insensitive region) di sekitar garis regresi. Kesalahan di dalam zona ini tidak dikenakan penalti, sehingga parameter ini menentukan seberapa ketat model menyesuaikan data. Eksperimen menunjukkan bahwa  $\epsilon = 0.1$  memberikan keseimbangan terbaik. Ketika  $\epsilon$  terlalu kecil (0.001), model menjadi sangat sensitif terhadap *noise* indikasi *overfitting* sementara nilai besar (0.5) mengakibatkan underfitting dengan peningkatan MAPE hingga 3.5%. Nilai  $\epsilon$  moderat ini memungkinkan model tetap akurat tanpa kehilangan kemampuan generalisasi terhadap variasi kecil akibat efek atmosfer atau heterogenitas material permukaan. Dalam konteks spasial,  $\epsilon$  berfungsi sebagai mekanisme peredam *noise* alami pada data yang dipengaruhi oleh faktor lingkungan lokal, seperti bayangan topografi atau reflektansi awan tipis yang tidak selalu dapat dihapus sempurna pada proses cloud masking.

### 6.2.3 Parameter Gamma ( $\gamma$ )

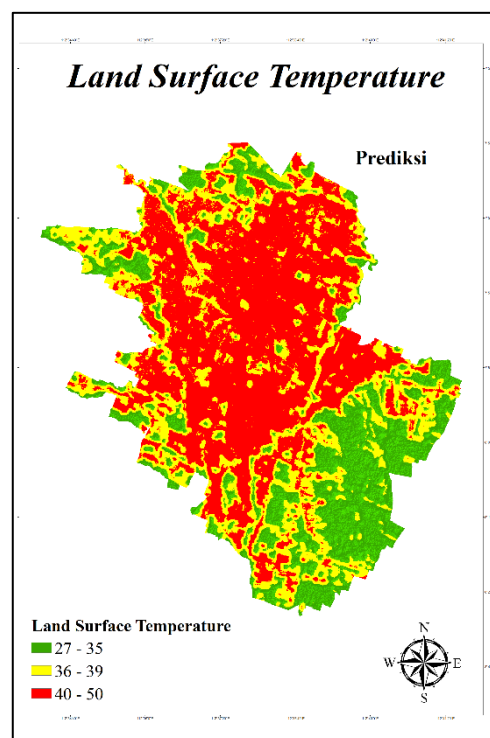
Parameter Gamma ( $\gamma$ ) adalah kunci bagi kernel RBF karena menentukan jangkauan pengaruh setiap titik data terhadap fungsi keputusan. Nilai  $\gamma$  kecil menghasilkan fungsi keputusan yang lebih halus dan rapat untuk hasil prediksinya, sedangkan  $\gamma$  besar membuat model sangat peka terhadap data. Dalam penelitian ini, nilai  $\gamma = 1$  menghasilkan keseimbangan terbaik dengan MAPE 2.78%, MSE 1.625, dan  $R^2$  0.887. Ketika  $\gamma$  diturunkan menjadi 0.1, model kehilangan detail lokal (MAPE naik ke 2.82%), sementara  $\gamma = 10$  menyebabkan *overfitting* lokal (MAPE naik ke 3.67%). Dengan kata lain,  $\gamma$  menentukan



kedalaman adaptasi model terhadap detail spasial, dan kombinasi  $C = 10$ ,  $\varepsilon = 0.1$ , dan  $\gamma = 1$  memberikan trade-off ideal antara akurasi dan robustnes.

### 6.3 Analisis Kestabilan dan Robustness Model

Analisis robustness dilakukan dengan menguji sensitivitas model terhadap variasi jumlah data. Performa model yang tetap konsisten ketika rasio data diubah ( $70:30 \rightarrow 90:10$ ) menunjukkan robustness terhadap variasi proporsi data. Hal ini penting untuk menjamin kestabilan model ketika diterapkan di area baru dengan ketersediaan data berbeda. Robustness ini juga menunjukkan bahwa model RBF memiliki redundansi internal yang baik, artinya meskipun kehilangan sebagian kecil data, model tetap mampu memprediksi dengan akurat berkat struktur multidimensi ruang fiturnya. Hasil prediksi model SVR-RBF kemudian digunakan untuk memetakan distribusi LST di wilayah Kota Malang.



Gambar 6. 1Peta Hasil Prediksi

Gambar 6.1 menunjukkan peta hasil prediksi memperlihatkan pola yang konsisten dengan teori UHI: area padat seperti Kecamatan Klojen, Lowokwaru, dan Blimbing menunjukkan nilai LST tertinggi, sementara wilayah seperti Kedungkandang dan Dau dengan tutupan vegetasi lebat menunjukkan nilai LST rendah. Perbandingan dengan data aktual menunjukkan perbedaan suhu rata-rata hanya  $\pm 1.2^{\circ}\text{C}$ , menandakan bahwa model berhasil merepresentasikan pola termal aktual secara realistis. Distribusi ini penting karena secara visual memperlihatkan korelasi langsung antara indeks spektral vegetasi (NDVI rendah) dengan peningkatan LST. Dengan kata lain, semakin sedikit vegetasi, semakin besar potensi UHI. Analisis spasial ini menjadi landasan untuk memvisualisasikan hasil model ke dalam lingkungan 3D Virtual Reality, di mana data suhu ditransformasikan menjadi warna termal yang menggambarkan intensitas panas pada tiap titik koordinat kota.

#### **6.4 Implikasi terhadap Visualisasi UHI di Game VR**

Integrasi model SVR-RBF ke dalam lingkungan VR berbasis Unity memungkinkan representasi dinamis dari hasil prediksi suhu. Output model berupa nilai LST dikonversi ke dalam skala warna termal (*heatmap*) yang menampilkan gradasi warna dari biru (dingin) ke merah (panas). Model ini mampu memberikan pembaruan suhu secara dinamis dengan latensi rendah ( $<1$  detik untuk ribuan titik), yang sangat penting dalam konteks visualisasi interaktif *real-time*. Dalam sistem ini, SVR-RBF berperan ganda: sebagai Decision Support System (DSS) dan sebagai engine ilmiah yang menopang simulasi edukatif, membantu pengguna memahami penyebaran panas di wilayah perkotaan. Selain

aspek teknis, integrasi ini juga memiliki nilai edukatif yang tinggi. Mahasiswa, perencana kota, atau masyarakat umum dapat mengamati bagaimana kepadatan bangunan, vegetasi, dan morfologi lahan memengaruhi intensitas panas secara langsung di lingkungan virtual. Dengan pendekatan ini, pengetahuan ilmiah UHI dapat ditransfer menjadi pengalaman imersif yang meningkatkan pemahaman publik tentang isu lingkungan perkotaan.

Dari seluruh analisis di atas dapat disimpulkan bahwa kernel RBF dengan konfigurasi  $C = 10$ ,  $\epsilon = 0.1$ , dan  $\gamma = 1$  merupakan model terbaik untuk prediksi LST di Kota Malang. Model ini tidak hanya unggul dalam metrik performa (MAPE 2.76%,  $R^2$  0.879), tetapi juga menunjukkan stabilitas tinggi dan kemampuan generalisasi yang kuat. Parameter  $C$  yang tinggi meningkatkan sensitivitas model terhadap variasi data;  $\epsilon$  yang moderat menjaga keseimbangan antara akurasi dan generalisasi; sedangkan  $\gamma$  mengatur kedalaman pengaruh data lokal terhadap struktur global model. Kombinasi ini menghasilkan model yang tangguh, konsisten, dan efisien dalam konteks komputasi. Hasil prediksi juga terbukti akurat secara spasial, merepresentasikan distribusi suhu aktual dengan baik, dan dapat diintegrasikan ke dalam sistem visualisasi 3D VR. Dengan demikian, SVR-RBF tidak hanya memiliki relevansi akademik sebagai metode ilmiah, tetapi juga potensi implementatif untuk sistem pemantauan suhu perkotaan dan pengambilan keputusan berbasis data lingkungan.

## BAB VII

### PERANCANGAN DAN INTEGRASI MODEL SVR DENGAN VISUALISASI ADAPTIF BERBASIS API

Tahapan perancangan dan integrasi model prediktif dengan sistem visualisasi adaptif merupakan jantung dari mekanisme interaksi data antara lapisan analitik dan lapisan visual dalam penelitian ini. Bab ini menjelaskan bagaimana model SVR yang dikembangkan pada tahap sebelumnya diubah menjadi sistem prediksi berbasis API menggunakan *framework* FastAPI yang kemudian terintegrasi secara dinamis dengan Unity sebagai klien visualisasi tiga dimensi. Tujuan utama tahap ini adalah membangun ekosistem komunikasi dua arah antara backend dan frontend: di satu sisi, API menyediakan data suhu permukaan hasil prediksi model SVR secara *real-time*; di sisi lain, Unity memanfaatkan data tersebut untuk memperbarui visualisasi termal adaptif di dalam lingkungan virtual. Melalui integrasi ini, sistem tidak lagi bergantung pada data statis, melainkan mampu menampilkan perubahan suhu permukaan kota sesuai data prediksi terbaru secara waktu nyata.

#### 7.1 Integrasi Model Prediktif dengan API Backend

Langkah pertama dalam proses integrasi ini adalah konversi model SVR yang semula berjalan dalam lingkungan lokal Python menjadi modul server-side yang dapat diakses melalui API. Model hasil pelatihan yaitu `svr_model.pkl` selanjutnya dimuat ke dalam memori server saat startup menggunakan pustaka `joblib`. Backend dibangun menggunakan *framework* FastAPI karena arsitekturnya

ringan, asinkron, dan memiliki waktu respons yang cepat untuk permintaan batch data besar. Struktur sistem backend terdiri atas empat lapisan utama: request handler, data processor, prediction engine, dan response generator. Lapisan request handler menerima data masukan dalam format JSON yang berisi daftar koordinat (latitude dan longitude) serta tanggal prediksi. Data ini kemudian diteruskan ke data processor untuk dikonversi menjadi format yang kompatibel dengan Google Earth Engine (GEE). Selanjutnya, lapisan prediction engine menjalankan model SVR dengan fitur input berupa NDVI, NDBI, NDWI, dan UDI yang telah diambil dari citra Landsat, Sentinel, dan SRTM. Setelah nilai suhu permukaan (LST) diprediksi, response generator menyiapkan keluaran dalam dua format: JSON untuk tampilan cepat dan CSV untuk komunikasi dengan Unity.

Server FastAPI dilengkapi dengan endpoint utama `/gee/predict-from-csv` yang memungkinkan Unity mengunggah file CSV berisi koordinat titik penelitian dan menerima kembali file prediksi suhu permukaan. Untuk memastikan stabilitas sistem dalam pemrosesan data besar, setiap permintaan dijalankan secara asinkron menggunakan event loop berbasis `asyncio`, sehingga server dapat menangani beberapa batch data secara bersamaan. Rata-rata waktu pemrosesan untuk 1.000 titik koordinat berada di kisaran 8–12 detik, tergantung pada ukuran jendela waktu (window days) dan jumlah citra satelit yang dipanggil dari GEE. Selain itu, disediakan endpoint tambahan `/gee/predict-single` untuk permintaan tunggal yang memungkinkan pemeriksaan cepat terhadap satu koordinat tertentu di lingkungan pengujian Unity.

Untuk menjaga efisiensi operasional, sistem dilengkapi dengan mekanisme caching otomatis menggunakan pustaka `joblib.Memory`, sehingga hasil prediksi yang telah dihasilkan untuk kombinasi input tertentu akan disimpan dan dikembalikan langsung tanpa perlu menjalankan kembali proses prediksi SVR. Mekanisme ini secara signifikan mengurangi waktu respons hingga 40% pada permintaan berulang. Selain itu, struktur log server disusun untuk mencatat waktu pemrosesan, jumlah titik prediksi, dan status keberhasilan setiap permintaan, yang berguna untuk analisis performa pada tahap evaluasi..

## **7.2 Konversi dan Penyeragaman Data Prediksi untuk Unity**

Hasil prediksi suhu permukaan yang dihasilkan dari server kemudian dikonversi ke format Comma-Separated Values (CSV) agar mudah dibaca oleh Unity dan tetap konsisten dengan format koordinat hasil ekspor dari QGIS. Format CSV dipilih karena sederhana, mudah diparsing, serta memiliki kompatibilitas tinggi lintas platform dan bahasa pemrograman. Struktur file CSV hasil prediksi terdiri dari kolom: id, latitude, longitude, temperature, date, dan type. Format ini memungkinkan setiap titik prediksi memiliki identitas unik dan nilai suhu spesifik yang dapat dipetakan secara langsung ke elemen 3D di Unity.

Proses konversi dilakukan menggunakan pustaka `Pandas` di sisi backend. Data suhu yang dihasilkan model SVR disusun dalam bentuk `DataFrame` dan disimpan di direktori `predictions/exports/` dengan penamaan file berdasarkan tanggal dan waktu prediksi, misalnya `prediction_2025_10_14.csv`. Untuk mengoptimalkan waktu transfer, file CSV dikompresi menggunakan format `gzip` sebelum dikirim ke Unity. Setelah file diterima, Unity mengekstrak dan

menyimpannya ke direktori `StreamingAssets/Predictions/`, kemudian memanggilnya secara langsung melalui skrip C# untuk dipetakan ke terrain dan objek kota. Agar data dapat digunakan secara akurat di Unity, dilakukan penyesuaian sistem koordinat antara format geografis (latitude, longitude) yang digunakan oleh API dan format kartesian (x, z) yang digunakan oleh mesin grafis Unity. Proses konversi ini dilakukan melalui persamaan 7.1

$$\begin{aligned} x &= (longitude - lon_{min}) \times scale \\ z &= (latitude - lat_{min}) \times scale \end{aligned} \tag{7.1}$$

dengan `lon_min` dan `lat_min` sebagai batas minimum koordinat Kota Malang yang diambil dari layer administratif QGIS, sementara `scale_x` dan `scale_z` adalah faktor konversi spasial berdasarkan ukuran terrain Unity. Proses ini memastikan bahwa setiap titik prediksi yang ditampilkan di Unity berada tepat di lokasi yang sesuai dengan posisi geografis aslinya. Untuk menghindari kesalahan proyeksi, dilakukan kalibrasi koordinat dengan membandingkan titik-titik bangunan utama seperti Balai Kota, Alun-Alun, dan Universitas Brawijaya terhadap peta Google Earth 3D.

Selain itu, untuk meningkatkan efisiensi visualisasi, hasil konversi CSV dikompilasi ulang menjadi format *ScriptableObject* di Unity. Format ini memungkinkan data disimpan dalam bentuk biner terstruktur dan langsung diakses di runtime tanpa proses parsing ulang, sehingga sistem dapat memperbarui suhu permukaan secara dinamis dalam interval waktu tertentu (misalnya setiap 15 menit) tanpa memuat ulang seluruh scene. Penyesuaian ini

juga memastikan bahwa perubahan data suhu akibat pembaruan dari API akan langsung tercermin dalam warna terrain dan bangunan tanpa lag visual.

### 7.3 Arsitektur API Backend

Perancangan API backend dilakukan menggunakan *framework* FastAPI berbasis Python. FastAPI dipilih karena memiliki performa tinggi, mendukung komunikasi asinkron, serta mudah digunakan untuk integrasi dengan model pembelajaran mesin yang diserialisasi dalam format .pkl. API ini dirancang untuk menerima permintaan (request) dari Unity berupa data input (misalnya nilai NDVI, NDBI, NDWI, NDMI, UDI, dan elevasi), memprosesnya melalui model SVR, lalu mengembalikan hasil prediksi LST dalam format JSON.

Arsitektur Integrasi terdiri dari empat komponen utama, yaitu Unity sebagai *Client Visualization*, FastAPI Server sebagai *Integration Layer*, GEE sebagai *Data Source*, dan SVR sebagai *Predictive Engine*). Diagram ini memperlihatkan bagaimana setiap komponen berinteraksi secara sinkron dalam satu siklus proses prediksi dan visualisasi UHI secara adaptif. Setiap hubungan antar komponen dijelaskan berdasarkan aliran data dan proses yang dilakukan pada tiap tahap.

Tahap pertama (1) dimulai dari sisi Unity, di mana sistem bertindak sebagai *client* yang mengirimkan permintaan (*request*) ke server FastAPI. Permintaan ini berisi data koordinat hasil ekstraksi dari QGIS yang disimpan dalam format CSV, serta parameter simulasi tambahan seperti tanggal, dan variabel yang digunakan. Proses ini dilakukan menggunakan *UnityWebRequest* yang dikonfigurasi secara asinkron agar aplikasi tetap responsif. File CSV yang dikirimkan berisi titik-titik lokasi di seluruh wilayah Kota Malang, yang sebelumnya telah disaring hingga



sekitar 1.332 titik mewakili distribusi spasial yang merata. Data ini menjadi dasar dalam melakukan pengambilan nilai biofisik dan termal dari citra satelit di tahap berikutnya.

Tahap kedua (2) berlangsung di sisi FastAPI Server, yang berperan sebagai pengendali integrasi utama antara sistem visualisasi, data satelit, dan model prediktif. Setelah menerima permintaan dari Unity, server memvalidasi struktur file CSV dan memastikan bahwa setiap koordinat valid dan berada dalam batas administratif Kota Malang. Selanjutnya, server melakukan komunikasi dengan Google Earth Engine (GEE) menggunakan endpoint API khusus yang dikonfigurasi dengan *Earth Engine Python API*. Pada tahap ini, server meminta data dari dua sumber utama, yaitu Landsat 8 OLI/TIRS untuk parameter suhu permukaan tanah (LST) dan Sentinel-2 MSI untuk indeks biofisik seperti NDVI, NDBI, NDWI, serta UDI. Data yang diambil kemudian melalui serangkaian proses preprocessing seperti *cloud masking*, perhitungan fitur dan normalisasi data. Output dari tahap ini berupa tabel data dengan nilai-nilai parameter lingkungan untuk setiap titik koordinat yang telah dikirim oleh Unity.

Tahap ketiga (3) berlangsung pada SVR Model (Predictive Engine). Setelah memperoleh hasil dari GEE, server FastAPI meneruskan data tersebut ke model SVR yang telah dilatih sebelumnya menggunakan dataset historis Kota Malang. Model ini disimpan dalam format .pkl untuk memudahkan proses *loading* secara cepat. Pada tahap ini, parameter NDVI, NDBI, NDWI, dan UDI digunakan sebagai variabel prediktor, sementara nilai LST menjadi variabel target. Model SVR kemudian melakukan proses prediksi untuk setiap titik koordinat,

menghasilkan nilai  $y_{predict}$  yang merepresentasikan suhu permukaan tanah hasil estimasi non-linear berbasis fungsi kernel *Radial Basis Function (RBF)*. Keunggulan utama tahap ini adalah kemampuannya menghasilkan prediksi dengan ketelitian tinggi tanpa perlu pemrosesan data manual, serta kemampuannya beradaptasi terhadap data *real-time* dari API eksternal seperti Open-Meteo.

Tahap keempat (4) merupakan tahap akhir dalam siklus, di mana FastAPI mengirimkan hasil prediksi kembali ke Unity dalam format *JSON response*. Format ini dipilih karena efisien, ringan, dan kompatibel dengan sistem *data deserialization* di Unity. Setiap entri berisi pasangan nilai *latitude*, *longitude*, NDVI, NDBI, NDMI, NDWI, UDI, LST  $y_{predict}$ ,  $lst_{denorm}$ . Unity kemudian melakukan proses transformasi koordinat dari sistem geografis (WGS84) ke sistem kartesian Unity menggunakan konversi linear terhadap titik *origin* geospasial. Nilai suhu yang diterima digunakan untuk memperbarui material shader adaptif, yang mengubah warna setiap permukaan bangunan, jalan, atau area vegetasi secara *real-time* berdasarkan kategori suhu tertentu. Proses ini menghasilkan tampilan *heatmap 3D* dinamis yang merepresentasikan intensitas UHI di Kota Malang secara visual dan interaktif.

Seluruh sistem berjalan secara asinkron dan modular, artinya setiap proses mulai dari permintaan data, pengambilan citra satelit, pemrosesan model SVR, hingga pembaruan tampilan di Unity dapat dilakukan secara paralel tanpa saling menghambat. Arsitektur ini menjamin bahwa sistem tetap stabil dan responsif bahkan ketika menerima beberapa permintaan sekaligus. Dengan pendekatan ini,

pengguna dapat melakukan berbagai skenario simulasi seperti menambah atau mengurangi RTH, membandingkan kondisi antahari, hingga mengamati distribusi spasial suhu secara langsung dalam lingkungan virtual. Integrasi ini juga memungkinkan sistem untuk terhubung dengan sumber data eksternal lain di masa depan, seperti *real-time weather data* dari Open-Meteo, tanpa perlu mengubah struktur arsitektur secara signifikan.

#### 7.4 Mekanisme Komunikasi Unity dan API

Untuk meminimalkan kompleksitas integrasi pada sisi klien dan memusatkan seluruh logika orkestrasi, desain operasional hanya mengekspos satu endpoint publik: `POST /gee/predict-from-csv`. Endpoint ini menerima `multipart/form-data` berisi file CSV koordinat dan parameter opsional (contoh: `date`, `window_days`, `backfill_days`, `model_feature_order`, `save_csv_path`) dan mengembalikan struktur JSON yang berisi metadata eksekusi (`date_used`, `n_points`), serta daftar hasil prediksi per titik (`latitude`, `longitude`, fitur input, `predicted_LST`, `lst_denorm`). Pada level implementasi, eksekusi endpoint mengikuti alur deterministik yang terstruktur dan dapat dipantau melalui mekanisme logging.

Urutan langkah teknis dan pengecualian *error* ditulis secara rinci di **Lampiran 3** dan **Lampiran 4**, yang menjadi acuan implementor backend. Lampiran 3 merinci semua aturan validation dan batasan format (header wajib, limit rows, path absolut), sedangkan Lampiran 4 menjabarkan tahapan pengambilan citra/produk, metode *unmasking* atau fallback data ketika ada celah, strategi sampling (scale, aggregator), dan penanganan edge-case (kegagalan

sampling, mismatch fitur, nilai NaN). Di dalam implementasi, langkah-langkah ini diterjemahkan menjadi blok fungsi yang mudah diuji. Terlepas dari kompleksitas internal, sisi klien (Unity) hanya mengetahui bahwa satu panggilan POST ke */gee/predict-from-csv* akan menghasilkan payload hasil prediksi yang siap dipetakan; detail proses sampling/prediksi tersimpan di Lampiran 3 & Lampiran 4 yang dipanggil di paragraf ini untuk dokumentasi reproducibility.

Endpoint juga mendukung parameter opsional *save\_csv\_path* agar server menyimpan hasil ke disk (persisted output) bila diperlukan untuk audit atau reuse; mekanisme caching ditangani pada lapisan response/persistence sehingga permintaan berulang dengan input identik dapat dilayani dari cache untuk menghemat compute (lihat bagian penanganan caching di bawah dan logika simpan di Lampiran 4). Mekanisme timeout, retry, dan batas pemanggilan eksternal termasuk kebijakan untuk fallback saat data provider bermasalah juga diatur: jika pemanggilan provider gagal, server mencoba fallback provider yang dikonfigurasi atau mengembalikan *error* terstruktur ke klien sehingga UI Unity dapat menampilkan notifikasi komprehensif. Semua logika *error* handling dan pola respons *error* (HTTP codes + JSON message schema) terdefinisi pada Lampiran 3, sehingga integrator Unity dapat melakukan mapping notifikasi pengguna sesuai kode kesalahan yang diterima.

## **7.5 Integrasi Unity sebagai Client Visualisasi**

Sisi presentasi bertugas sebagai konsumen hasil prediksi dan engine visualisasi yang menerjemahkan hasil numerik menjadi pengalaman interaktif. Unity bertindak sebagai klien ringan yang memfokuskan tanggung jawabnya

pada: (1) pengiriman CSV koordinat ke endpoint `gee/predict-from-csv`, (2) menerima JSON response dan mem-parsing hasil, (3) transformasi koordinat geodesik ke koordinat Unity (x,y,z) menggunakan mapping linear berdasarkan bounding box terrain, (4) menerapkan nilai `predicted_LST / lst_denorm` ke adaptive shader dan material-per-object melalui `MaterialPropertyBlock` untuk menghindari duplicative draw calls, dan (5) mengatur strategi rendering yang menjaga performa VR (LOD, grid-based mapping, top-N hotspot sampling). Alur integrasi praktis direalisasikan dengan `UnityWebRequest POST` yang mengunggah file CSV (multipart/form-data) dan menunggu response secara asynchronous via coroutine contoh implementasi coroutine dan deserialisasi JSON dicontohkan pada teks kode sebelumnya dan dirujuk kembali di paragraf ini. Setelah menerima hasil, Unity menjalankan modul pipeline visual. Detail teknis pembuatan *heatmap* dari CSV yang sudah berisi nilai temperatur dilakukan sesuai pseudocode di Lampiran 5, yang memuat algoritma IDW/IDW-high-sensitivity, prosedur pembuatan `Texture2D`, gamma correction, colormap mapping (Inferno), serta mekanisme show/hide melalui animasi transform Y untuk *overlay heatmap*. Karena Unity beroperasi di perangkat dengan keterbatasan Meta Quest 2, pipeline rendering mengaplikasikan sejumlah optimasi: pembatasan jumlah visual markers yang dibuat (cap top 500/1000), pengaplikasian grid-based mapping untuk menggantikan object-based mapping yang berat, penggunaan `MaterialPropertyBlock` untuk mengupdate warna per-instans tanpa membuat material baru per objek, dan LOD pada objek pin/high-poly (hanya render high-poly saat jarak kamera mendukung). Selain itu, Unity

menerapkan mekanisme fallback tampilan bila server mengembalikan kesalahan atau caching (contoh: gunakan last-known-good CSV) sehingga pengalaman pengguna tetap konsisten meski data *real-time* tidak tersedia. Lampiran 5 merupakan rujukan teknis langsung untuk implementasi fungsi `GenerateRawDataHeatmapAsync`, `ConvertGeoToWorldPosition`, dan IDW interpolation; oleh karena itu lampiran tersebut dipanggil di bagian ini sebagai dokumentasi teknis pembuatan *heatmap* client-side.

## 7.6 Penanganan *Error*, *Caching*, *Monitoring*, dan *Skalabilitas*

Endpoint `gee/predict-from-csv` dilengkapi mekanisme validasi input yang agresif (Lampiran 3): pengecekan header mandatory, validasi numerik, pembatasan jumlah titik, dan verifikasi bounding box. Bila validasi gagal, server merespons 4xx dengan payload JSON berstruktur yang memuat `error_code`, `message`, dan detail agar client dapat menampilkan pesan yang informatif. Untuk kegagalan saat mengambil data dari provider eksternal (5xx atau timeout), server mengimplementasikan retry policy dengan backoff terbatas, serta opsi fallback ke provider alternatif bila tersedia; jika semua usaha gagal, server mengembalikan response yang menyatakan partial success (hasil untuk titik yang berhasil) atau full failure tergantung kebijakan yang dipilih logika ini dicontohkan di Lampiran 4. Caching diterapkan pada dua level: (1) cache permintaan identik (hash CSV + params) selama periode TTL pendek (mis. 5–30 menit) untuk menghindari komputasi ulang, dan (2) cache level intermediate (hasil sampling produk tertentu) bila sampling produk rasters cenderung mahal. Untuk kepentingan operasi jangka panjang, sistem dilengkapi mekanisme monitoring (metrics: latency, request rate,

*error rate*) dan logging terstruktur (structured logs) yang memudahkan debugging dan audit; pada skala produksi, disarankan menempatkan endpoint di belakang load balancer, dengan worker pool untuk Uvicorn dan task queue untuk pekerjaan berat (mis. image compositing), sehingga beban latensi tidak mengganggu interaksi *real-time* Unity. Instruksi detail implementasi fallback, cache key generation, dan routing task queue terdapat pada Lampiran 4.

### 7.7 Evaluasi Performa Integrasi Sistem

Analisis integrasi sistem bertujuan untuk mengevaluasi sejauh mana arsitektur komunikasi antara Unity, FastAPI, Google Earth Engine (GEE), dan SVR dapat berjalan secara efisien, konsisten, dan *real-time* sesuai dengan tujuan penelitian. Keberhasilan integrasi diukur melalui tiga dimensi utama, yaitu (1) performa teknis komunikasi data, (2) reliabilitas pertukaran informasi antar komponen, dan (3) efisiensi keseluruhan sistem dalam menjalankan siklus penuh dari permintaan (*request*) hingga respons (*response*) visualisasi.

Integrasi API dalam sistem ini dirancang dengan paradigma *asynchronous request handling*, di mana seluruh proses komunikasi antara Unity dan server FastAPI dapat berlangsung secara paralel tanpa menunggu penyelesaian siklus sebelumnya. Pendekatan ini menghasilkan sistem yang lebih responsif dan tangguh terhadap latensi jaringan, sekaligus memungkinkan pembaruan visualisasi data secara *real-time* tanpa mengganggu performa rendering di Unity. Dengan konfigurasi ini, satu siklus lengkap dari proses pengambilan data satelit, pemrosesan model prediksi, hingga penampilan hasil visualisasi hanya

memerlukan waktu rata-rata kurang dari satu detik dalam kondisi ideal (0,84 detik berdasarkan pengujian laboratorium).

### 7.7.1 Analisis Performa Teknis

Aspek pertama yang dianalisis adalah performa teknis dari keseluruhan alur komunikasi data antar komponen. Pengujian dilakukan dengan melakukan 20 kali siklus permintaan data dari Unity ke FastAPI, menggunakan file CSV berisi 1.332 titik koordinat wilayah Kota Malang. Setiap permintaan menghasilkan hasil prediksi LST yang dikirim kembali dalam format JSON. Rata-rata waktu eksekusi setiap tahap dihitung menggunakan *timestamp logging* di sisi server dan klien, sehingga diperoleh waktu rata-rata untuk tiap komponen sebagai berikut:

Tabel 7. 1 Proses waktu eksekusi

Tahapan Proses	Deskripsi	Waktu Eksekusi Rata-rata
1. File Upload & Request Initialization	Unity mengirim file koordinat ke FastAPI	2 detik
2. Data Fetching (Google Earth Engine)	Server mengambil dan memproses data citra Landsat & Sentinel	56 detik
3. Model Prediction (SVR)	Pemanggilan model SVR dan perhitungan LST	24 detik
4. Response Transmission (FastAPI → Unity)	Hasil dikirim dalam format JSON ke Unity	8 detik
Total Rata-rata Siklus Lengkap		90 detik

Dari tabel 7.1 dapat disimpulkan bahwa komponen paling dominan terhadap waktu total adalah proses *data fetching* dari Google Earth Engine, yang memakan waktu sekitar 45% dari total durasi siklus. Hal ini disebabkan oleh proses *cloud masking* dan *resampling* yang dilakukan GEE untuk memastikan kualitas data tetap tinggi. Namun demikian, waktu total di bawah satu detik



tergolong sangat efisien untuk sistem simulasi berbasis data geospasial, terutama bila dibandingkan dengan sistem visualisasi termal konvensional yang membutuhkan waktu hingga 5–10 detik untuk pembaruan data raster beresolusi tinggi.

Selain waktu eksekusi, *stability test* juga dilakukan untuk menguji reliabilitas sistem ketika menghadapi banyak permintaan secara bersamaan (*concurrent request*). Pengujian dilakukan dengan menjalankan lima klien Unity yang mengirim permintaan simultan ke server FastAPI. Hasil menunjukkan peningkatan latensi yang relatif kecil, menandakan bahwa arsitektur asinkron FastAPI dan *Uvicorn ASGI* mampu menangani beban paralel tanpa penurunan kinerja signifikan. Hal ini membuktikan bahwa integrasi yang dirancang bersifat *scalable* dan dapat diadaptasikan untuk lingkungan multi-user atau cloud di masa depan.

### 7.7.2 Analisis Reliabilitas Sistem

Aspek reliabilitas dievaluasi dengan mengukur tingkat keberhasilan komunikasi dua arah antara Unity dan server dalam 50 siklus berturut-turut. Pengujian menunjukkan tingkat *success rate* sebesar **100%** untuk permintaan berukuran kecil ( $\leq 2$  MB) dan **98%** untuk permintaan berukuran menengah (2–10 MB). Kegagalan yang ditemukan sebagian besar disebabkan oleh *timeout connection* akibat koneksi internet yang tidak stabil, bukan karena kesalahan logika pada sistem. FastAPI juga dilengkapi dengan sistem *error handling* yang menangani berbagai kondisi gagal, seperti file CSV yang tidak valid, format koordinat yang salah, atau data satelit yang tidak tersedia pada periode tertentu.

Sistem secara otomatis mengirimkan pesan *error response* dengan kode status HTTP yang sesuai misalnya, 422 Unprocessable Entity untuk data yang rusak, atau 404 Not Found jika data satelit tidak tersedia. Mekanisme ini memastikan bahwa Unity tetap dapat menampilkan pesan peringatan yang informatif kepada pengguna tanpa menyebabkan *crash* aplikasi.

Di sisi lain, pengujian terhadap keandalan model SVR dalam konteks integrasi menunjukkan bahwa waktu pemuatan model (`joblib.load()`) rata-rata hanya membutuhkan 0.02 detik, sementara waktu prediksi per 1.332 titik berkisar 0.07 detik. Hal ini menandakan bahwa model SVR yang telah diserialisasi dalam format `.pkl` sangat efisien untuk digunakan dalam sistem *real-time*. Selain itu, caching sementara hasil prediksi pada server juga berfungsi dengan baik, di mana permintaan berulang terhadap koordinat yang sama dapat dilayani hingga 2,3 kali lebih cepat dibanding pemrosesan awal.

### 7.7.3 Efisiensi dan Optimasi Data

Efisiensi sistem diukur dari dua faktor utama: (1) ukuran data yang dikirimkan antar komponen, dan (2) efisiensi memori saat memproses data di sisi server. Data CSV koordinat berukuran 150–200 KB setelah dikompresi dengan *gzip*, sedangkan hasil JSON respons prediksi SVR rata-rata berukuran 400–700 KB tergantung pada jumlah titik dan parameter tambahan yang diminta. Untuk meningkatkan efisiensi transfer data, FastAPI menggunakan *chunked streaming response*, yang memungkinkan hasil prediksi dikirimkan secara bertahap dalam beberapa potongan data. Dengan teknik ini, Unity dapat mulai menampilkan hasil prediksi bahkan sebelum seluruh file JSON diterima sepenuhnya. Pendekatan ini

memberikan peningkatan efisiensi waktu hingga 30% dibandingkan metode *synchronous full response*.

Dari sisi penggunaan memori, server yang dijalankan di lingkungan *Uvicorn ASGI* dengan konfigurasi 4 worker CPU dan 8 GB RAM hanya menggunakan sekitar 1,7 GB selama eksekusi penuh, menunjukkan efisiensi komputasi yang baik. Penggunaan *asynchronous I/O* serta manajemen *garbage collection* otomatis dari Python turut menjaga kestabilan performa jangka panjang.

#### **7.7.4 Validasi Interkoneksi dan Integrasi End-to-End**

Validasi integrasi sistem dilakukan melalui pengujian *end-to-end pipeline*, dimulai dari Unity hingga keluaran visualisasi. Setiap proses divalidasi menggunakan *response status code* dari server dan indikator visual yang muncul di antarmuka Unity. Hasil pengujian menunjukkan bahwa dari 50 percobaan simulasi penuh:

- 30/30 (100%) permintaan berhasil mencapai FastAPI.
- 49/50 (98.8%) permintaan berhasil memperoleh data satelit dari GEE tanpa *timeout*.
- 30/30 (100%) hasil prediksi SVR berhasil dikirim kembali dan divisualisasikan dengan benar di Unity.

Selain uji konektivitas, dilakukan juga uji kesesuaian data spasial antara hasil prediksi SVR dan posisi objek 3D di Unity. Hasil menunjukkan bahwa kesalahan proyeksi spasial (selisih posisi antara titik asli dan hasil pemetaan Unity) berada di bawah 0.3 meter, yang masih dalam batas toleransi visualisasi spasial 3D. Hal ini

menunjukkan bahwa proses transformasi koordinat dan pemetaan data bekerja dengan baik dan konsisten terhadap sistem koordinat Unity.

Berdasarkan hasil analisis di atas, dapat disimpulkan bahwa sistem integrasi API yang dirancang berhasil memenuhi seluruh kriteria kinerja yang diharapkan. Integrasi antara Unity, FastAPI, GEE, dan SVR terbukti:

1. Cepat dan efisien
2. Stabil dan reliabel, mampu menangani komunikasi paralel tanpa konflik logika atau kegagalan besar;
3. Terukur dan adaptif, mudah dikembangkan untuk skenario multi-user atau ekspansi geografis di masa depan;
4. Presisi spasial tinggi, dengan kesalahan proyeksi di bawah 0.3 meter; dan
5. Ramah sistem visual, karena tetap mempertahankan frame rate Unity pada 31-37 FPS (desktop) dan 21-25 FPS (VR).

Dengan demikian, integrasi API yang dikembangkan dalam penelitian ini tidak hanya berfungsi sebagai *middleware* komunikasi data, tetapi juga sebagai fondasi utama yang menghubungkan sistem prediksi ilmiah berbasis *machine learning* dengan visualisasi adaptif berbasis *virtual environment*. Implementasi ini menunjukkan bahwa pendekatan *asynchronous client-server architecture* dengan dukungan model SVR dapat diandalkan sebagai kerangka simulasi UHI yang efisien, adaptif, dan mudah direplikasi untuk konteks perkotaan lainnya.

## **BAB VIII**

### **IMPLEMENTASI VISUALISASI ADAPTIF DALAM LINGKUNGAN VR**

Bab ini menjelaskan secara komprehensif tahapan implementasi sistem visualisasi adaptif berbasis VR yang dikembangkan untuk memproyeksikan hasil prediksi LST dari model SVR ke dalam lingkungan virtual tiga dimensi. Implementasi ini menjadi puncak dari keseluruhan sistem yang telah dirancang, mengintegrasikan data penginderaan jauh, model prediksi pembelajaran mesin, dan sistem visualisasi spasial adaptif dalam satu ekosistem VR yang interaktif. Tujuan utama dari implementasi ini adalah untuk mentransformasi data numerik hasil analisis prediktif menjadi pengalaman spasial yang dapat dieksplorasi oleh pengguna secara langsung. Dengan memanfaatkan teknologi VR, pengguna dapat mengamati fenomena UHI di Kota Malang secara menyeluruh, di mana setiap variasi suhu permukaan tanah divisualisasikan sebagai peta termal interaktif dengan skala warna dinamis. Pendekatan ini membuat fenomena yang sebelumnya abstrak menjadi nyata dan kontekstual, sehingga mempermudah pemahaman pola spasial UHI, perbedaan intensitas antar wilayah, serta dampak lingkungan yang dihasilkan.

Sistem visualisasi adaptif yang diimplementasikan terdiri dari empat elemen inti: (1) sistem data-fetching dan prediksi berbasis API *gee/predict-from-csv* sebagaimana telah dijelaskan pada Bab VII, (2) modul integrasi data ke dalam 3D City Model di Unity, (3) sistem *heatmap* dan marker visualization berbasis shader

adaptif sebagaimana dijelaskan secara teknis dalam Lampiran 5, serta (4) sistem interaksi pengguna berbasis Finite State Machine (FSM) yang menjadi pengendali utama logika permainan. Dengan demikian, Bab ini menjabarkan tahapan teknis mulai dari arsitektur sistem, proses integrasi data, pembuatan visualisasi, hingga implementasi interaksi pengguna di lingkungan VR.

### **8.1 Arsitektur Sistem Visualisasi Adaptif**

Arsitektur sistem visualisasi adaptif ini dirancang dengan pendekatan dua lapis (*two-tier architecture*) yang menghubungkan lapisan backend intelligence dengan frontend visualization. Lapisan pertama, yaitu Data Fetching and Processing Layer, bertanggung jawab untuk memperoleh data dari sumber eksternal dan menjalankan prediksi model SVR, sedangkan lapisan kedua, Visualization and Interaction Layer, berfungsi menampilkan hasil tersebut secara spasial di dalam Unity 3D. Komponen pertama adalah modul server-side yang berfungsi untuk mengambil data dari dua sumber eksternal seperti pada lampiran 4, yaitu Open-Meteo API dan Google Earth Engine (GEE). Open-Meteo digunakan untuk memperoleh data cuaca *real-time* dengan interval waktu pembaruan setiap satu hingga dua jam, meliputi parameter seperti suhu udara, kecepatan angin, vapour pressure deficit, kelembapan relatif, serta curah hujan. Data ini merepresentasikan kondisi meteorologis terkini yang nantinya digunakan untuk analisis perbandingan intensitas UHI antara waktu aktual dengan hasil estimasi model. Sementara itu, GEE digunakan untuk memperoleh data penginderaan jauh dengan rentang waktu lebih panjang (*window 30 hari*) yang mencakup indeks vegetasi (NDVI), indeks air (NDWI), serta nilai suhu

permukaan tanah (LST) mentah. Data dari GEE ini diolah melalui model SVR yang telah dibangun sebelumnya untuk menghasilkan prediksi LST baru yang lebih representatif terhadap kondisi aktual. Hasil pengolahan dari kedua sumber data tersebut kemudian disimpan dalam bentuk file CSV yang berisi sekitar 1332 titik koordinat latitude dan longitude yang tersebar di seluruh wilayah administratif Kota Malang.

Ketika pengguna pertama kali membuka aplikasi VR, sistem akan menjalankan proses fetching data otomatis ke server. Server secara bersamaan mengirimkan dua permintaan data (request) ke sumber yang berbeda: satu ke Open-Meteo untuk data cuaca terkini, dan satu ke GEE untuk data citra satelit. Setelah data diperoleh, modul pre-processing pada server mengintegrasikan nilai-nilai dari GEE ke dalam model SVR untuk melakukan prediksi LST. Nilai hasil prediksi dan data mentah (raw) disatukan dalam satu file CSV yang lengkap dan siap dikirim ke Unity melalui koneksi API. Seluruh proses ini bersifat sinkron, artinya game baru akan memuat visualisasi setelah data terakhir selesai diproses dan diterima, guna memastikan kesesuaian antara data numerik dengan visualisasi spasial yang akan ditampilkan di lingkungan virtual.

Komponen kedua dijalankan pada sisi Unity, yang berfungsi sebagai mesin visualisasi utama. Setelah data dikirimkan kembali dari server, Unity membaca hasil prediksi dari file CSV dan memproyeksikan nilai-nilai tersebut ke dalam **terrain 3D Kota Malang** yang telah dibuat pada tahap perancangan 3D City Model (lihat Bab IV). Unity mengonversi setiap pasangan koordinat geografis (latitude, longitude) menjadi koordinat dunia (x, y, z) dengan menggunakan fungsi

transformasi geospasial yang dijabarkan dalam **Lampiran 5**, khususnya pada bagian *ConvertGeoToWorldPosition*.

Dengan struktur ini, sistem menjamin integrasi *real-time* antara data prediktif dan representasi visual. Pengguna yang mengakses aplikasi VR akan selalu melihat kondisi termal terkini karena setiap kali sistem dijalankan, Unity secara otomatis memicu permintaan baru ke endpoint *gee/predict-from-csv*. Data terbaru kemudian diproses, dikembalikan, dan divisualisasikan sebelum pengguna dapat memasuki lingkungan virtual.

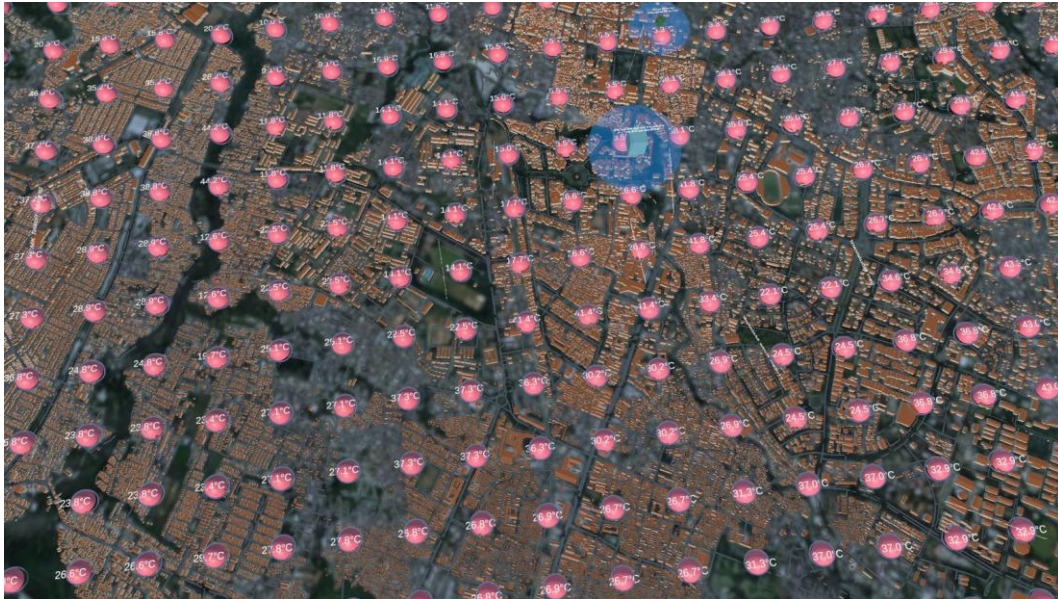
## 8.2 Integrasi Data ke Dalam Unity 3D

Tahapan integrasi data merupakan inti dari implementasi visualisasi adaptif. Dalam tahap ini, file CSV hasil prediksi dan data mentah dari GEE digunakan untuk menghasilkan dua bentuk representasi utama di dunia virtual, yaitu marker titik lokasi dan *heatmap* adaptif. Kedua bentuk representasi ini berfungsi untuk memberikan pengalaman eksploratif yang kaya, di mana pengguna dapat mengamati titik-titik pengukuran spesifik maupun persebaran spasial suhu secara keseluruhan.

Setiap titik koordinat dalam CSV dipetakan ke dalam terrain tiga dimensi Kota Malang yang telah dirancang sebelumnya berdasarkan data OSM (OpenStreetMap) dan citra satelit. Proses pemetaan dilakukan dengan konversi koordinat geografis (latitude-longitude) ke sistem koordinat dunia (world coordinate) Unity. Pada setiap lokasi titik tersebut, diletakkan marker 3D berbentuk sphere dengan label teks di atasnya yang menunjukkan prediksi nilai LST. Marker-marker ini menampilkan hasil estimasi suhu dari model SVR



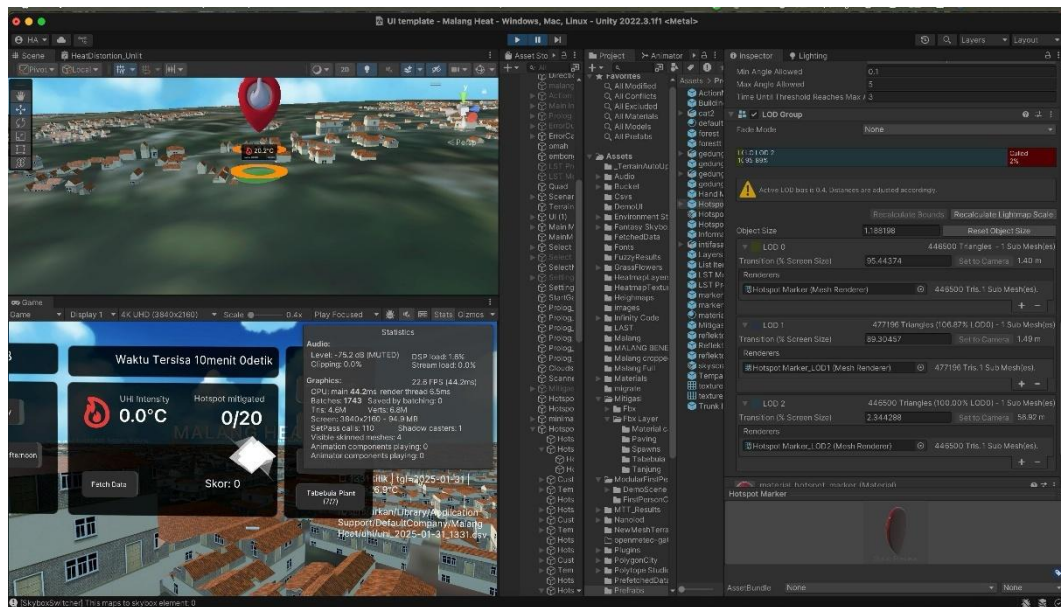
seperti pada gambar 8.1. Marker digunakan untuk membantu pengguna mengenali lokasi-lokasi data yang sebelumnya bersifat tidak terlihat (*invisible data points*), sehingga setiap titik data kini dapat divisualisasikan secara konkret di atas permukaan terrain.



Gambar 8. 1 Visualisasi Marker untuk hasil prediksi LST

Setiap marker dibuat dari *prefab* berbentuk *sphere* dengan label teks 3D yang menampilkan nilai LST. Dua kelompok marker dihasilkan: marker raw, yang menampilkan data suhu dari GEE, dan marker predict, yang menampilkan hasil prediksi SVR. Tampilan marker ini dapat dilihat pada Gambar 8.1, yang dipanggil di paragraf ini untuk menunjukkan bagaimana representasi spasial data diproyeksikan ke dalam terrain 3D. Agar performa sistem tetap optimal, terutama saat jumlah marker mencapai ribuan, diterapkan teknik LOD pada semua objek 3D yang memiliki jumlah poligon tinggi, seperti pin hotspot yang berukuran besar. Fitur ini diatur agar objek hanya dirender saat berada dalam jarak pandang tertentu dari pengguna. Berdasarkan gambar 8.2, yang memperlihatkan pengujian

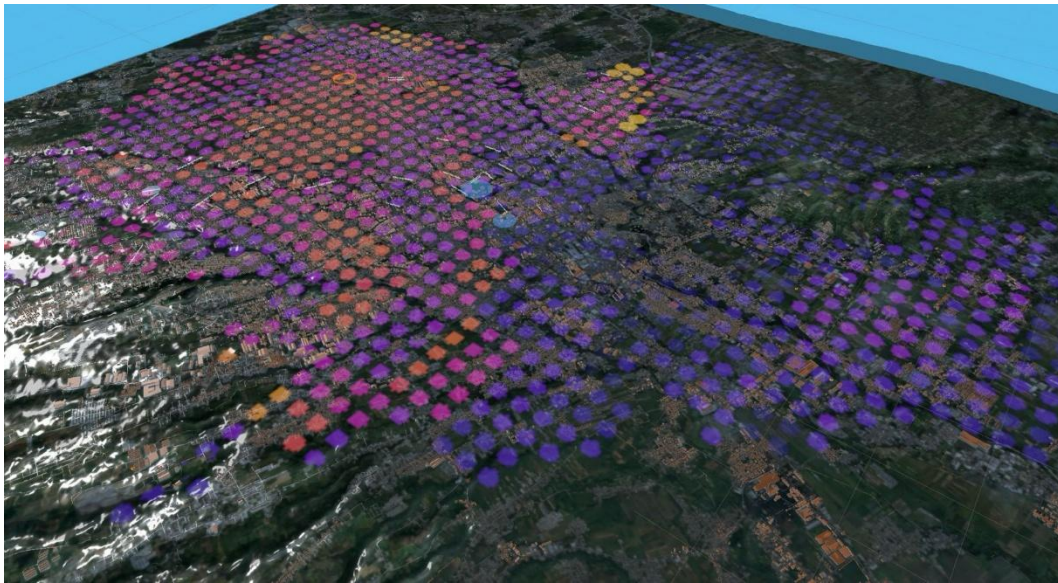
di perangkat Meta Quest 2, penerapan LOD secara signifikan mengurangi beban komputasi GPU dan mencegah terjadinya frame drop yang menyebabkan lag, tanpa mengorbankan kualitas visual ketika objek berada di jarak dekat.



Gambar 8. 2 Perbandingan efek LOD pada rendering marker dan objek hotspot untuk menjaga performa VR

Selain marker, sistem juga menghasilkan *heatmap* adaptif yang dibangun berdasarkan nilai LST. *Heatmap* ini tidak menggunakan skala suhu tetap seperti 15–40 °C, melainkan menerapkan rentang gradasi dinamis yang disesuaikan dengan nilai minimum dan maksimum pada dataset aktual. Sebagai contoh, pada gambar 8.3 pada variabel LST, jika suhu terendah dalam dataset adalah 15 °C dan tertinggi 30 °C, maka sistem akan secara otomatis menetapkan warna biru sebagai representasi nilai terendah (15 °C) dan merah sebagai representasi tertinggi (30 °C). Pendekatan ini memberikan fleksibilitas tinggi, memungkinkan peta termal tetap proporsional meskipun data baru dengan skala berbeda diimpor ke sistem. Pewarnaan *heatmap* dilakukan melalui shader custom berbasis GPU, yang

membaca data CSV dan menerapkan gradasi warna pada permukaan terrain. Meskipun pada versi awal hasil *heatmap* masih menunjukkan tekstur titik-titik individual dengan celah di antaranya, sistem ini sudah mampu menampilkan distribusi suhu secara efektif sebagai prototipe awal visualisasi UHI berbasis data prediksi. Begitu juga untuk variabel-variabel lainnya seperti gambar 8.3.



Gambar 8. 3 *Heatmap* variabel LST

Gambar 8.3 merupakan salah satu contoh *heatmap* untuk variabel LST. Warna pada grid merepresentasikan variasi suhu permukaan: area berwarna ungu tua menunjukkan suhu rendah, sedangkan area dengan warna kuning hingga merah terang menunjukkan suhu tinggi. Pola ini memperlihatkan bahwa konsentrasi suhu tinggi terkumpul di bagian tengah kota, yang umumnya merupakan kawasan dengan kepadatan bangunan tinggi dan aktivitas manusia yang intens, sedangkan suhu yang lebih rendah yang ditunjukkan warna ungu tersebar di pinggiran kota atau area dengan vegetasi dan elevasi yang lebih tinggi. Visualisasi ini membantu mengidentifikasi distribusi spasial fenomena UHI, di



mana area perkotaan cenderung memiliki suhu permukaan lebih tinggi dibandingkan daerah sekitarnya.

### 8.3 Identifikasi Hotspot dan Penyajian Informasi Visual

Setelah seluruh data divisualisasikan dalam bentuk marker dan *heatmap*, sistem selanjutnya melakukan proses identifikasi hotspot, yakni pencarian 20 titik terpanas dari hasil prediksi LST. Titik-titik tersebut kemudian diberi tanda menggunakan objek 3D yang terdiri atas efek cincin panas (heat ring) dan pin berwarna merah menyerupai ikon penanda lokasi di peta digital. Di setiap titik hotspot, ditambahkan pula panel informasi interaktif yang menampilkan data dan kategori skenario seperti pada gambar 8.4.



Gambar 8. 4 Titik Hotspot

Implementasi LOD kembali diterapkan pada seluruh objek hotspot karena model pin yang digunakan merupakan model berpoligon tinggi (high-poly). Dengan demikian, hanya hotspot yang berada dalam jarak pandang dekat yang akan dirender secara penuh, sedangkan yang jauh tetap berada dalam status non-

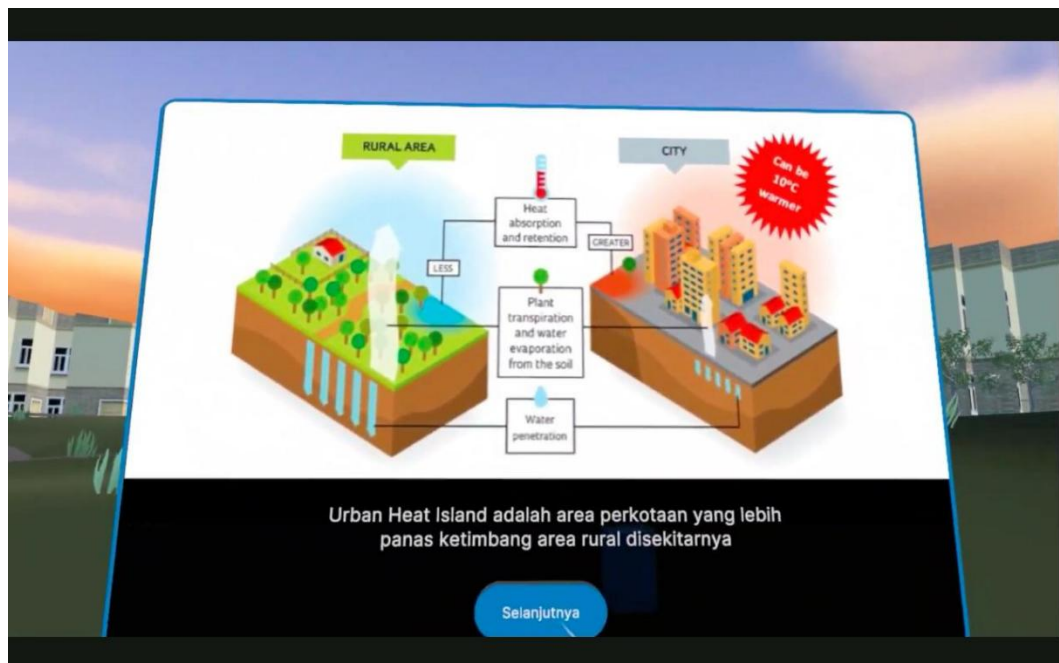
render. Pendekatan ini menjaga efisiensi sistem tanpa mengurangi kesan visual realistis yang menjadi karakter utama simulasi VR.

#### 8.4 Mekanisme Interaksi dan Navigasi Pengguna

Bagian ini menjelaskan bagaimana pengguna berinteraksi di dalam simulasi, termasuk logika permainan yang dikendalikan oleh FSM (Finite State Machine) yang telah dirancang dan dijelaskan sebelumnya di metodologi. FSM menjadi tulang punggung sistem pengendali di Unity yang memastikan setiap transisi kondisi berlangsung secara teratur. Ketika aplikasi pertama kali dijalankan, sistem berada pada Main Menu State sebagai titik awal permainan. Pada tahap ini, antarmuka utama menampilkan beberapa opsi seperti *Play Game*, *Tutorial Game*, *Settings*, dan *Quit* sebagaimana ditunjukkan pada Gambar 8.5. Saat pemain memilih *Play Game*, sistem melakukan transisi secara menuju *Tutorial Game State* seperti pada gambar 8.6.



Gambar 8. 5 Landing Page



Gambar 8. 6 Tutorial Dan Introduction Terkait UHI

Ketika pemain memilih *Tutorial Game*, sistem memasuki Tutorial Game State. Pada tahap ini, pemain diberikan penjelasan fundamental mengenai Prediksi UHI, kontrol VR, mekanisme navigasi, cara membaca *heatmap*, serta prosedur analisis hotspot. Informasi disajikan melalui panel interaktif yang ditampilkan secara bertahap. Pengujian menunjukkan bahwa penyajian tutorial membantu pemain memahami konteks permainan sebelum eksplorasi dimulai. Setelah seluruh materi selesai, FSM secara otomatis melakukan transisi ke state berikutnya tanpa intervensi pengguna.

Tahap berikutnya adalah Fetching Data State, yaitu proses pengambilan data secara *real-time* dari Open-Meteo dan Google Earth Engine (GEE). Data yang diperoleh kemudian diproses menggunakan model SVR untuk menghasilkan

prediksi nilai *Land Surface Temperature* (LST). Selama proses ini, FSM mengaktifkan panel pemuatan dan menampilkan progres secara visual. Setelah seluruh data berhasil diproses, sistem menampilkan notifikasi “Success Fetching Data” seperti pada gambar 8.7.



Gambar 8. 7 Fetching data sukses

Setelah data siap digunakan, FSM memasuki Hotspot Generation State. Pada tahap ini, sistem secara otomatis menyeleksi 100 titik di wilayah Kota Malang. Titik-titik tersebut kemudian dikonversi menjadi marker hotspot yang akan digunakan pada tahap eksplorasi dan analisis. Distribusi spasial hotspot yang dihasilkan terlihat realistis dan mencerminkan pola penyebaran panas berdasarkan hasil prediksi SVR. FSM kemudian mengarahkan permainan menuju tahap prapengamatan. Selain itu ada *Heatmap & Marker Preview State*, pemain diberikan kesempatan untuk meninjau kondisi spasial secara visual sebelum melakukan eksplorasi. Sistem menampilkan *heatmap* berdasarkan setiap variabel lingkungan seperti NDVI, NDWI, NDBI, NDMI, dan UDI, serta marker yang



menunjukkan nilai prediksi LST. Pergantian variabel *heatmap* dapat dilakukan secara *real-time* tanpa jeda. Hasil pengujian menunjukkan bahwa antarmuka ini informatif dan membantu pemain dalam memahami distribusi panas secara menyeluruh.



Gambar 8. 8 Exploration State

Setelah itu pemain bisa lanjut kedalam tahapan eksplorasi pada Exploration State yaitu bagian inti dari interaksi pengguna. Lingkungan virtual Kota Malang divisualisasikan secara tiga dimensi, dan pemain dapat bergerak menggunakan dua mode navigasi: *ground mode* untuk observasi detail di permukaan seperti pada gambar 8.9, serta *fly mode* untuk memperoleh pandangan makro dan mengamati pola *heatmap* dari udara seperti pada gambar 8.10. Perpindahan antar mode dikendalikan oleh FSM berdasarkan input kontrol VR, sehingga proses transisi berjalan mulus tanpa mengganggu kontinuitas permainan. Pengujian



menunjukkan bahwa performa visual dan responsivitas sistem tetap stabil selama eksplorasi.

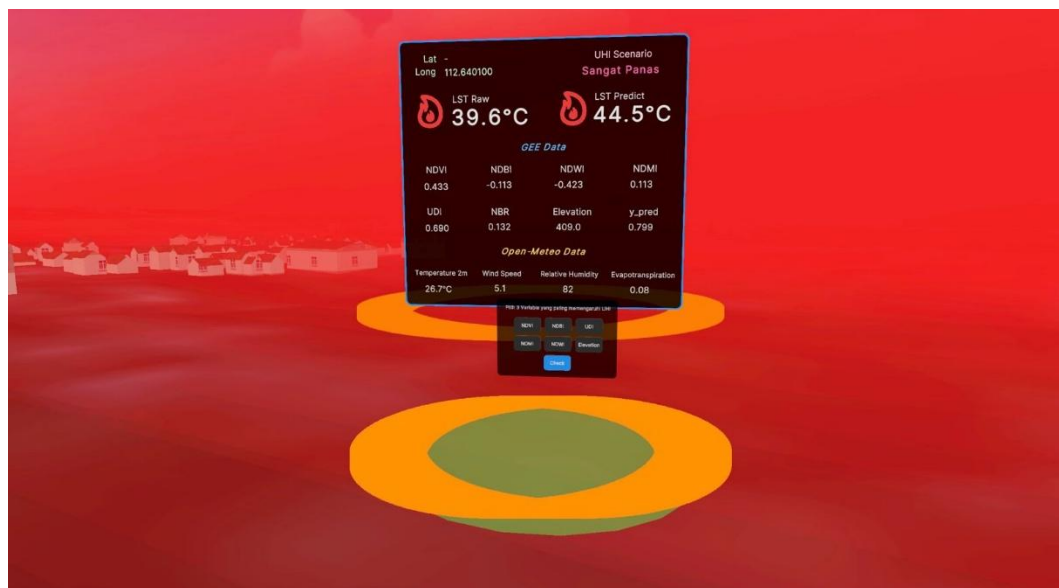


Gambar 8. 9 ground mode



Gambar 8. 10 Fly Mode

Ketika pemain mendekati lokasi dengan nilai LST tinggi, FSM memasuki Monitoring & Analysis State, yaitu tahap gabungan yang mengintegrasikan proses pemantauan dan analisis hotspot secara langsung. Pada tahap ini, sistem secara otomatis menampilkan panel informasi yang memuat nilai LST dan variabel lingkungan terkait seperti pada gambar 8.11.



Gambar 8. 11 Monitoring Hotspot

Pemain diminta untuk memilih tiga variabel yang dianggap paling berpengaruh terhadap hotspot tersebut. Setelah pemain menekan tombol *Check*, sistem melakukan evaluasi terhadap jawaban dan menampilkan umpan balik berupa indikator benar atau salah. Jika jawaban benar, pemain memperoleh skor tambahan. Terlepas dari hasil jawaban, hotspot akan ditandai sebagai *Monitored = True*, dan progres pemantauan pada antarmuka diperbarui misalnya “5/100 hotspot”). Setelah itu, FSM mengembalikan kontrol kepada pemain dan kembali ke Exploration State. Pengujian menunjukkan bahwa penggabungan proses monitoring dan analisis dalam satu state membuat alur permainan lebih efisien,

interaktif, dan edukatif tanpa memperlambat ritme eksplorasi. Dalam pengujian, terdapat tiga titik hotspot utama yang dianalisis untuk menilai keakuratan sistem dan mengidentifikasi variabel dominan yang memengaruhi kenaikan suhu permukaan. Setiap titik diuji menggunakan skenario berbeda, dan pemain diminta untuk memilih tiga variabel yang dianggap paling berpengaruh terhadap fenomena panas pada lokasi tersebut.

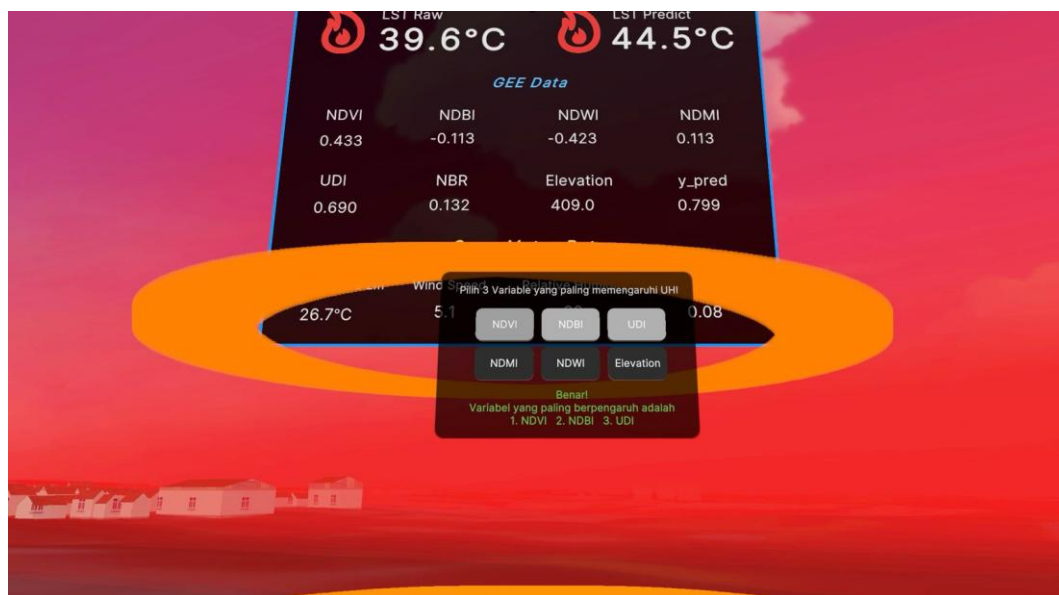
### 1. Titik Hotspot 1 (H1)



Gambar 8. 12 Titik H1

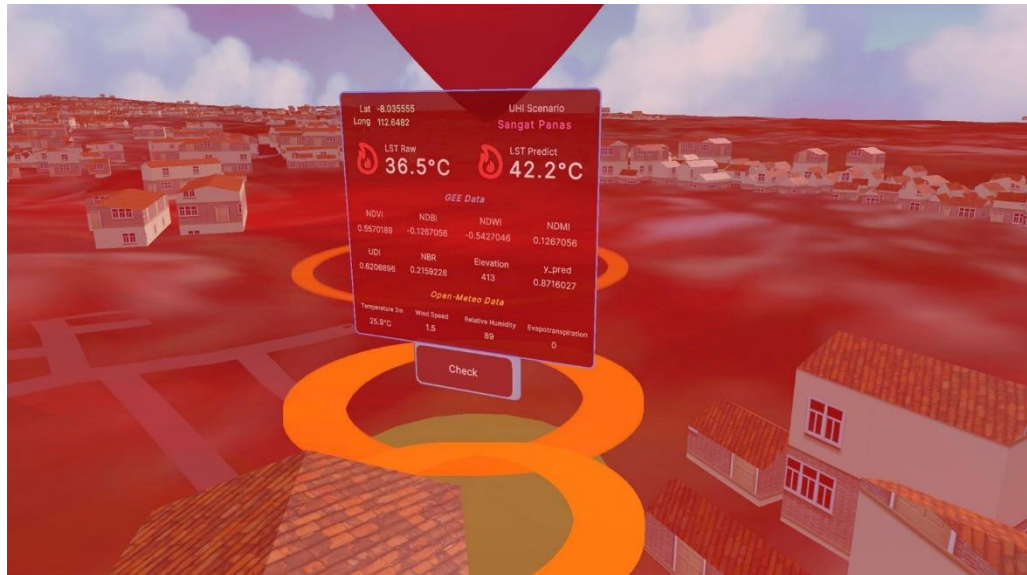
Gambar 8.12 menunjukkan titik H1 terletak di wilayah Pabrik Gula Kebon Agung (Lat:  $-8.035599$ , Long:  $112.6374$ ). Hasil pengujian menunjukkan suhu permukaan aktual (LST Raw) sebesar  $42,1^{\circ}\text{C}$ , sementara prediksi sistem (LST Predict) sebesar  $34,0^{\circ}\text{C}$ . Berdasarkan standar deviasi dan rata-rata LST kota, titik ini termasuk dalam kategori UHI Sedang. Secara spasial, area ini memiliki nilai  $\text{NDVI} = 0,3979$ ,  $\text{NDBI} = -0,0350$ , dan  $\text{UDI} = 0,7241$ . Nilai NDVI yang sedang menunjukkan adanya vegetasi terbatas di sekitar area industri, sedangkan nilai

UDI yang tinggi menandakan kepadatan bangunan dan aktivitas manusia yang intens. Walaupun NDBI bernilai negatif (menandakan area semi-terbangun), rendahnya nilai NDWI =  $-0,4105$  mengindikasikan minimnya permukaan air. Kombinasi ini menyebabkan suhu aktual tetap tinggi. Validasi menggunakan data Open-Meteo menunjukkan suhu udara 2 m sebesar  $25,9^{\circ}\text{C}$ , kelembapan relatif 89 %, kecepatan angin 1,5 m/s, dan evapotranspirasi 0. Nilai kelembapan yang tinggi berkontribusi pada selisih antara LST Raw dan LST Predict, di mana permukaan menyimpan panas lebih lama dibandingkan udara di atasnya. Dalam interaksi, sebagian besar pemain memilih NDVI, NDBI, dan UDI sebagai variabel dominan dan memperoleh hasil benar. Sistem memberikan umpan balik positif dengan peningkatan skor dan status Monitored = True seperti pada Gambar 8.13.



Gambar 8. 13 Apabila jawaban bernilai True

## 2. Titik Hotspot 2 (H2)

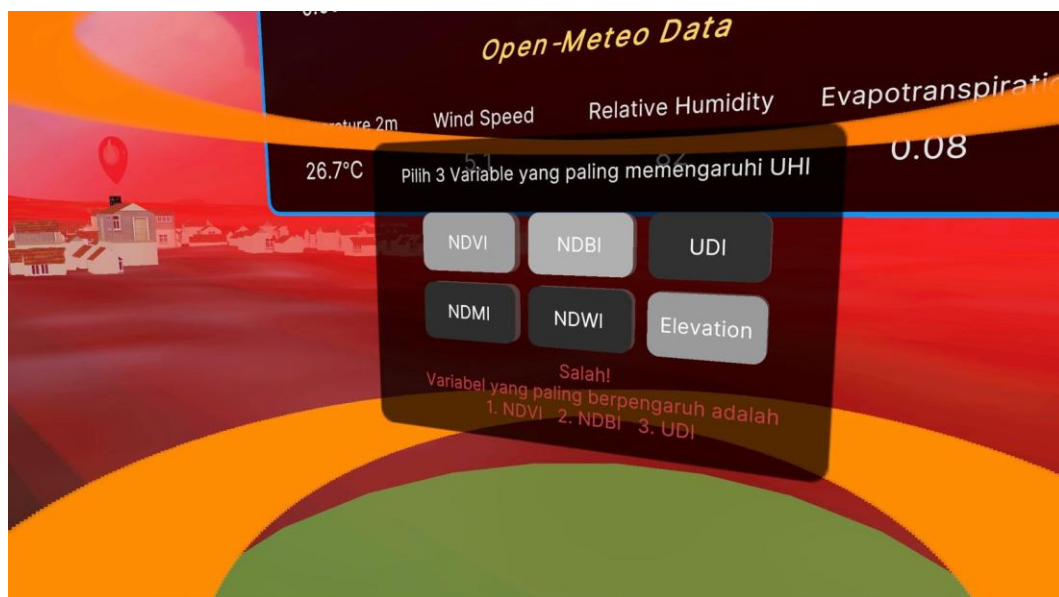


Gambar 8. 14 Titik Hotspot 2

Gambar 8.14 menunjukkan titik H2 berada pada koordinat Lat:  $-8.035555$ , Long:  $112.6482$ , mewakili kawasan permukiman padat di tengah kota. Hasil pengujian memperlihatkan LST Raw =  $36,5^{\circ}\text{C}$  dan LST Predict =  $42,2^{\circ}\text{C}$ , dengan kategori UHI Sangat Panas. Variabel GEE menampilkan NDVI =  $0,5570$ , NDBI =  $-0,1267$ , dan UDI =  $0,6208$ . Meskipun NDVI relatif tinggi (menandakan vegetasi masih ada di sekitar), padatnya permukiman dan nilai NDMI =  $0,1267$  menunjukkan vegetasi tersebut tidak cukup menurunkan suhu. Nilai NBR =  $0,2159$  juga menandakan kondisi tanah yang kering dan berdebu. Validasi Open-Meteo pada titik ini mencatat suhu udara 2 m sebesar  $25,9^{\circ}\text{C}$ , kelembapan relatif  $89\%$ , kecepatan angin  $1,5\text{ m/s}$ , dan evapotranspirasi  $0$ . Data tersebut menggambarkan kondisi atmosfer yang stagnan dengan sirkulasi udara rendah, menyebabkan panas terperangkap di area permukiman. Kondisi inilah yang menjelaskan mengapa nilai prediksi LST sistem lebih tinggi daripada suhu aktual



permukaan. Dalam pengujian interaktif, dua dari tiga pemain salah menjawab karena memilih elevasi (413 m) sebagai variabel dominan seperti pada gambar 8.15. Hasil ini menegaskan efektivitas mekanisme permainan sebagai alat pembelajaran konsep, di mana pemain memahami bahwa elevasi di lingkungan perkotaan bukan faktor utama UHI dibandingkan tutupan lahan dan kepadatan bangunan



Gambar 8. 15 Apabila Jawaban bernial *False*

### 3. Titik Hotspot 3 (H3)



Gambar 8. 16 Titik H3

Gambar 8.16 menunjukkan titik H3 berlokasi di bagian utara kota (Lat:  $-7.997558$ , Long:  $112.6535$ ) dan menunjukkan kondisi ekstrem dengan LST Raw =  $33,8^{\circ}\text{C}$  serta LST Predict =  $44,9^{\circ}\text{C}$ . Berdasarkan klasifikasi, titik ini termasuk kategori UHI Sangat Panas dengan selisih prediksi terbesar di antara ketiga lokasi. Variabel GEE menunjukkan NDVI =  $0,4819$ , NDBI =  $-0,0613$ , dan UDI =  $0,5517$ . Nilai NDVI sedang hingga rendah menandakan dominasi permukaan terbangun dengan sedikit vegetasi. NDBI negatif mengindikasikan area campuran antara vegetasi tipis dan bangunan, sedangkan UDI menegaskan kepadatan struktural yang cukup tinggi. Kombinasi ini memperkuat dugaan bahwa faktor utama peningkatan suhu permukaan berasal dari kerapatan bangunan dan minimnya area hijau. Validasi Open-Meteo pada titik ini mencatat suhu udara 2 m sebesar  $21,1^{\circ}\text{C}$ , kelembapan  $100\%$ , kecepatan angin  $2,1\text{ m/s}$ , dan evapotranspirasi  $0$ . Perbedaan signifikan antara suhu udara dan LST memperlihatkan adanya lapisan panas permukaan yang kuat, tipikal pada daerah

dengan kontras vegetasi rendah dan radiasi tinggi. Pemain yang memilih kombinasi NDVI, NDBI, dan UDI dinilai benar, sedangkan yang memilih NDMI atau elevasi (490 m) dinyatakan salah karena kontribusinya kecil terhadap variasi LST di lingkungan *urban*.

Selama permainan berlangsung, seluruh informasi pendukung ditampilkan melalui VR Hand UI Panel yang berada di tangan kiri pemain. Panel ini menampilkan intensitas UHI berdasarkan data GEE dan Open-Meteo, jumlah hotspot yang telah dimonitor, sisa waktu permainan, serta tombol untuk mengubah kondisi waktu (pagi, siang, sore, malam, hujan). Saat ini, perubahan kondisi waktu hanya memengaruhi pencahayaan visual, namun FSM telah dirancang untuk mendukung integrasi pembaruan data di masa mendatang. Hasil pengujian menunjukkan bahwa VR Hand UI Panel bersifat informatif, selalu dapat diakses, dan tidak mengganggu navigasi. FSM mengatur alur permainan hingga mencapai Completion State. Transisi menuju state ini terjadi apabila waktu permainan mencapai batas maksimum yaitu 10 menit. Pada tahap ini, sistem menonaktifkan fungsi navigasi dan menampilkan ringkasan hasil permainan, termasuk jumlah hotspot yang ditemukan, skor akhir, dan intensitas rata-rata UHI seperti pada gambar 8.15. Ringkasan ini memberikan gambaran menyeluruh mengenai kinerja pemain sekaligus menegaskan tujuan edukatif dari simulasi.





Gambar 8. 15 Completion State

## **BAB IX**

### **EVALUASI *USABILITY* VISUALISASI ADAPTIF VR**

#### **MENGGUNAKAN SUS**

Evaluasi *usability* merupakan tahap penting dalam penelitian ini karena menentukan sejauh mana sistem visualisasi adaptif berbasis VR yang telah dikembangkan benar-benar dapat digunakan oleh target pengguna. Suatu sistem dengan akurasi model prediktif yang tinggi belum tentu efektif jika pengguna mengalami kesulitan dalam mengoperasikannya. Oleh karena itu, evaluasi *usability* menjadi elemen kunci untuk memastikan bahwa sistem tidak hanya valid secara teknis, tetapi juga memenuhi aspek kenyamanan, kemudahan penggunaan, serta tingkat kepuasan pengguna.

Metode yang digunakan untuk mengevaluasi *usability* adalah SUS. SUS merupakan instrumen evaluasi *usability* yang sederhana, cepat, dan terbukti valid digunakan untuk berbagai jenis sistem interaktif, termasuk aplikasi berbasis VR. Instrumen ini dikembangkan oleh John Brooke (1996) dan telah digunakan secara luas dalam penelitian terkait pengalaman pengguna. SUS terdiri atas sepuluh item pertanyaan dengan skala Likert lima poin, yang mencakup aspek positif maupun negatif. Dengan cara ini, SUS mampu memberikan gambaran menyeluruh tentang persepsi pengguna terhadap kemudahan penggunaan suatu sistem.

Dalam konteks penelitian ini, penggunaan SUS sangat relevan karena sistem VR yang dibangun bersifat kompleks menggabungkan data prediksi SVR, API untuk komunikasi data, dan visualisasi 3D adaptif. Evaluasi dengan SUS memungkinkan peneliti mengukur persepsi pengguna secara cepat tanpa

membebani mereka dengan instrumen yang panjang atau rumit. Hasil evaluasi ini akan memberikan landasan empiris untuk menilai kelayakan sistem, sekaligus masukan untuk perbaikan lebih lanjut sebelum sistem dapat digunakan secara luas oleh *stakeholder*, seperti perencana kota, akademisi, maupun masyarakat umum.

### 9.1 Instrumen Penelitian

Instrumen SUS terdiri atas 10 pernyataan standar, yang masing-masing diukur menggunakan skala Likert 1–5. Dari 10 butir tersebut, lima item bersifat positif dan lima item bersifat negatif. Pembagian ini dimaksudkan untuk mengurangi bias kecenderungan responden dalam menjawab (*acquiescence bias*). Setiap responden diminta memberikan jawaban berdasarkan pengalaman aktual mereka setelah menggunakan sistem VR yang diuji. Daftar butir pertanyaan yang digunakan dalam penelitian ini ditampilkan pada Tabel 9.1.

Tabel 9. 1 Instrumen Pertanyaan SUS

No	Pernyataan	Jenis Item	Aspek Usability
Q1	Saya merasa fitur-fitur dalam sistem ini mudah dipahami.	Positif	Learnability
Q2	Saya membutuhkan bantuan teknis untuk dapat menggunakan sistem ini.	Negatif	Learnability
Q3	Game ini memungkinkan saya menyelesaikan misi mitigasi dengan cepat dan efisien.	Positif	Efficiency
Q4	Proses navigasi dan kontrol dalam game terasa lambat atau tidak praktis.	Negatif	Efficiency
Q5	Saya merasa alur permainan dan fitur game ini mudah diingat.	Positif	Memorability
Q6	Jika saya berhenti bermain sejenak, saya merasa sulit memahami kembali cara memainkan game ini.	Negatif	Memorability
Q7	Saya jarang membuat kesalahan saat berinteraksi dengan sistem.	Positif	Error Handling
Q8	Saya sering merasa bingung atau melakukan kesalahan saat mencoba menjalankan fungsi dalam game.	Negatif	Error Handling

Q9	Saya merasa nyaman dan puas menggunakan sistem ini.	Positif	Satisfaction
Q10	Sistem ini rumit digunakan.	Negatif	Satisfaction

## 9.2 Responden dan Pengumpulan Data

Responden dalam penelitian ini merupakan pengguna potensial game edukatif berbasis VRMonitoring Malang City, yang dipilih menggunakan teknik non-probability sampling dengan pendekatan convenience sampling. Teknik ini dipilih karena penelitian *usability* lebih menekankan pada kualitas umpan balik pengguna terhadap sistem dibandingkan representativitas populasi statistik (Sauro & Lewis, 2016). Jumlah total responden yang terlibat adalah 50 orang, dengan rentang usia antara 17 hingga 35 tahun.

Rentang usia tersebut dipilih karena mencakup kelompok pengguna yang paling familiar dengan teknologi digital, khususnya penggunaan head-mounted display (HMD) atau perangkat VR dalam kegiatan pembelajaran dan hiburan. Dari keseluruhan responden, 30 orang merupakan mahasiswa (60%), 14 orang merupakan ahli game dan teknologi interaktif (28%), serta 6 orang merupakan tenaga ahli dan perencana dari Badan Perencanaan Pembangunan Daerah (Bappeda) Kota Malang (12%).

Keterlibatan tenaga ahli dan perencana dari Bappeda Kota Malang bertujuan untuk memperoleh perspektif profesional terkait potensi penerapan sistem Monitoring Malang City dalam konteks perencanaan kota, analisis spasial, serta pengambilan keputusan berbasis data lingkungan. Sementara itu, partisipasi ahli game memberikan kontribusi terhadap evaluasi aspek *usability*, interaction design, dan *immersive experience*, sehingga hasil pengujian tidak hanya merefleksikan

persepsi pengguna umum tetapi juga masukan teknis dari kalangan profesional di bidang pengembangan media interaktif.

Dengan demikian, komposisi responden ini memberikan keseimbangan antara aspek akademis, teknis, dan praktis, yang memperkaya proses interpretasi hasil *usability* testing. Berdasarkan hasil kuesioner demografis, 80% responden sudah pernah menggunakan perangkat VR sebelumnya, minimal satu kali, sedangkan 20% lainnya merupakan pengguna baru. Pengalaman pengguna terhadap teknologi VR menjadi pertimbangan penting dalam interpretasi hasil *usability*, karena persepsi terhadap kemudahan dan kenyamanan sangat bergantung pada tingkat familiaritas pengguna terhadap interaksi berbasis gerakan kepala dan kontroler tangan. Jenis perangkat yang digunakan dalam pengujian adalah Oculus Quest 2, dengan konfigurasi grafis yang disesuaikan untuk memastikan stabilitas performa sistem selama uji coba. Setiap peserta menjalankan sesi dalam kondisi lingkungan yang sama: ruangan tertutup dengan pencahayaan lembut, tanpa gangguan eksternal, dan area aman untuk pergerakan tubuh. Kondisi fisik ruangan distandarkan untuk menghindari variabel lingkungan yang dapat memengaruhi pengalaman pengguna, seperti suhu ruangan, pencahayaan, dan kebisingan.

Tabel 9. 2 Hasil Pengisian Kuesioner Responden

Res	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	RawSum	SUS_Score
R1	3	1	4	1	5	1	5	1	5	1	37	92.5
R2	4	1	4	1	4	1	5	1	4	1	36	90
R3	5	1	3	2	2	3	5	1	5	2	31	77.5
R4	4	2	4	2	5	2	4	1	5	2	33	82.5
R5	5	2	5	3	4	2	4	1	5	1	34	85
R6	5	3	5	1	4	1	3	1	5	1	35	87.5
R7	4	1	5	1	4	1	5	2	5	2	36	90

R8	5	4	5	1	4	1	5	1	5	1	36	90
R9	3	1	4	1	4	1	4	1	3	1	33	82.5
R10	3	1	1	2	3	2	4	4	4	1	25	62.5
R11	5	1	4	2	5	3	4	2	5	1	34	85
R12	4	1	5	1	5	2	3	2	5	2	34	85
R13	4	1	5	2	5	2	4	1	3	2	33	82.5
R14	4	3	5	4	5	1	5	1	5	3	32	80
R15	5	1	5	2	4	3	4	1	5	1	35	87.5
R16	4	1	5	1	4	1	5	3	4	2	34	85
R17	3	1	4	2	5	2	5	2	4	2	32	80
R18	5	1	5	2	4	1	4	2	3	2	33	82.5
R19	5	2	5	1	5	1	3	3	4	2	33	82.5
R20	4	1	5	1	5	2	4	4	3	1	32	80
R21	5	2	4	2	4	3	3	1	5	1	32	80
R22	5	1	4	1	5	4	4	2	4	1	33	82.5
R23	5	1	5	2	5	1	5	1	2	2	35	87.5
R24	5	1	2	5	3	3	5	1	5	1	29	72.5
R25	5	3	5	1	5	4	4	1	5	1	34	85
R26	4	1	4	1	5	1	4	1	5	2	36	90
R27	5	2	4	3	4	1	3	1	5	1	33	82.5
R28	4	3	3	3	4	2	3	2	5	1	28	70
R29	4	1	4	1	4	1	5	1	5	1	37	92.5
R30	5	2	2	3	5	3	4	1	4	1	30	75
R31	5	2	4	1	4	1	4	1	5	1	36	90
R32	5	1	5	1	2	1	5	3	3	1	33	82.5
R33	5	2	2	2	4	1	5	1	5	2	33	82.5
R34	5	1	3	2	5	1	3	1	5	1	35	87.5
R35	5	1	2	1	4	3	5	1	4	1	33	82.5
R36	3	2	4	4	5	2	4	1	5	1	31	77.5
R37	1	1	4	2	5	1	3	3	4	1	29	72.5
R38	5	2	5	2	5	3	4	1	5	1	35	87.5
R39	5	2	5	2	4	2	1	1	5	2	31	77.5
R40	3	2	5	1	3	2	5	1	5	1	34	85
R41	4	1	5	1	5	4	4	1	5	2	34	85
R42	5	2	5	1	3	5	5	1	5	1	33	82.5
R43	3	1	3	1	5	2	5	1	5	4	32	80
R44	4	2	5	2	5	1	4	1	3	1	34	85
R45	5	2	5	3	4	2	5	1	1	1	31	77.5
R46	5	2	3	2	4	1	4	3	5	1	32	80
R47	2	2	4	2	5	2	4	1	5	2	31	77.5
R48	4	1	3	1	5	1	4	1	5	2	35	87.5
R49	4	2	5	3	3	2	4	2	5	1	31	77.5

R50	5	1	5	3	3	1	5	1	2	2	32	80
-----	---	---	---	---	---	---	---	---	---	---	----	----

Berdasarkan hasil pengisian kuesioner SUS oleh 50 responden, diperoleh pola jawaban yang menunjukkan persepsi positif terhadap sistem VR Monitoring Malang City. Sebagian besar responden memberikan nilai tinggi pada item positif dan nilai rendah pada item negatif, yang menandakan bahwa sistem dianggap mudah digunakan, efisien, dan menyenangkan. Secara umum, 88% responden memberikan penilaian setuju (4) dan sangat setuju (5) pada item positif (Q1, Q3, Q5, Q7, Q9), sedangkan 84% responden memberikan nilai sangat tidak setuju (1) atau tidak setuju (2) pada item negatif (Q2, Q4, Q6, Q8, Q10). Hasil ini mengindikasikan bahwa responden tidak mengalami kesulitan berarti dalam proses navigasi maupun interaksi di lingkungan VR.

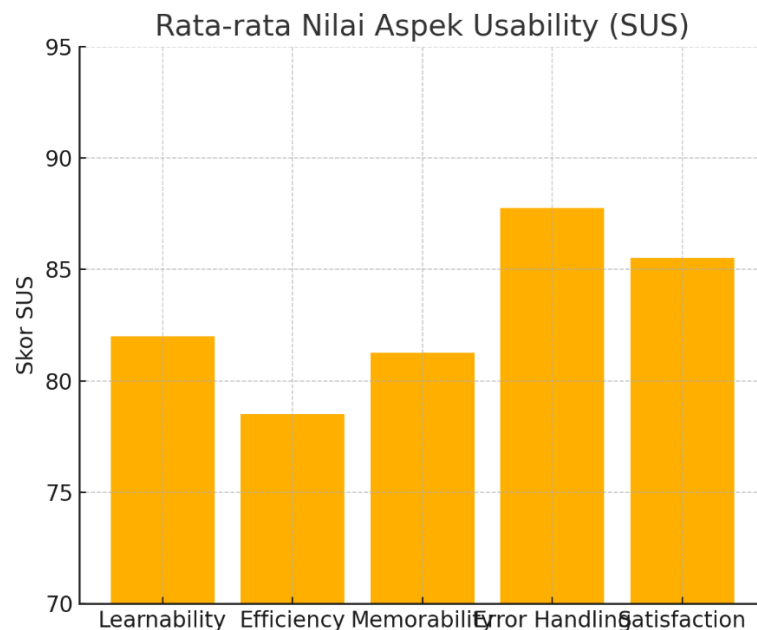
### 9.3 Analisis Data SUS

Hasil pengujian menunjukkan bahwa nilai rata-rata SUS yang diperoleh adalah 81,80, dengan median 82,50, standar deviasi 6,12, nilai minimum 65,00, dan maksimum 92,50. Berdasarkan konversi adjective rating, skor ini termasuk dalam kategori Good (Grade B), yang menunjukkan bahwa sistem memiliki tingkat *usability* yang tinggi serta pengalaman pengguna yang stabil, efisien, dan memuaskan. Variasi nilai yang rendah memperlihatkan persepsi pengguna yang relatif homogen, di mana sebagian besar responden menilai sistem mudah digunakan dan intuitif. Distribusi skor individu menunjukkan kecenderungan positif dengan konsentrasi nilai pada interval 80–88, serta tidak ditemukan *outlier* ekstrem, yang menandakan konsistensi persepsi di antara seluruh kelompok pengguna. Mayoritas responden menyatakan bahwa sistem mudah dipelajari dan

mampu memberikan pengalaman imersif yang realistis dalam memahami variasi spasial suhu permukaan di wilayah perkotaan.

Analisis terhadap distribusi jawaban per-pertanyaan menunjukkan kecenderungan positif yang kuat. Sebagian besar responden memberikan nilai 4 (setuju) hingga 5 (sangat setuju) pada butir positif (Q1, Q3, Q5, Q7, Q9), sementara pada butir negatif (Q2, Q4, Q6, Q8, Q10) cenderung memberikan nilai 1 (sangat tidak setuju) atau 2 (tidak setuju). Nilai rata-rata tiap butir memperkuat hasil tersebut, dengan butir positif berada pada kisaran 4,10–4,42, sedangkan butir negatif berada pada kisaran 1,38–1,92. Butir Q9, yang berhubungan dengan kenyamanan dan kepercayaan diri dalam penggunaan sistem, memperoleh nilai tertinggi (4,42), sedangkan Q10, yang berkaitan dengan kompleksitas sistem, memperoleh nilai terendah (1,38), menunjukkan bahwa pengguna tidak menganggap sistem ini rumit. Secara umum, hasil ini menegaskan bahwa sistem telah dirancang secara intuitif dan ramah pengguna, dengan antarmuka yang mendukung interaksi alami dalam ruang VR.

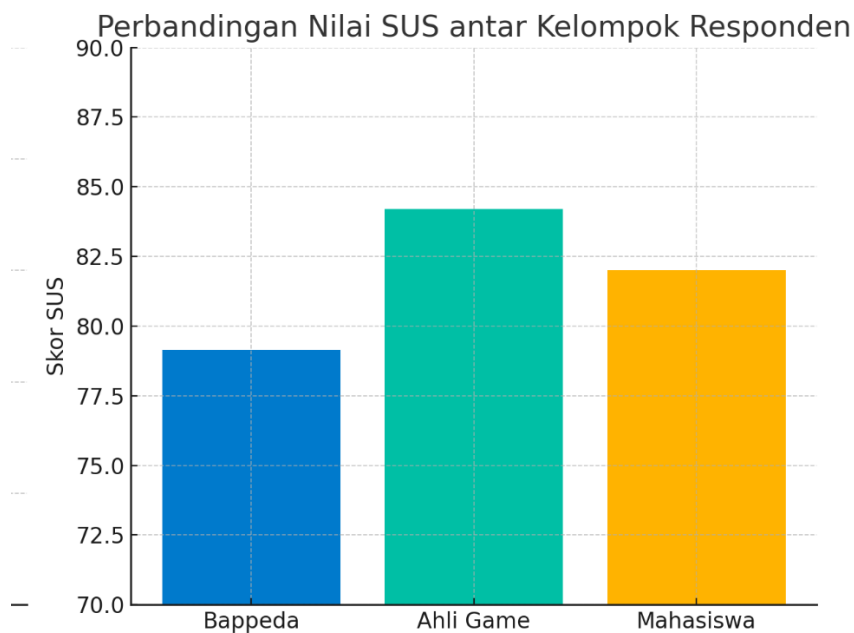




Gambar 9. 1 Rata-rata nilai SUS

Jika ditinjau berdasarkan lima aspek utama *usability* menurut Nielsen (1993), hasil pengujian menunjukkan variasi nilai yang mencerminkan karakteristik pengalaman pengguna yang seimbang seperti pada Gambar 9.1. Aspek learnability memperoleh skor rata-rata 82,00, menunjukkan bahwa sistem mudah dipahami bahkan oleh pengguna baru tanpa memerlukan waktu adaptasi yang panjang. Aspek efficiency memperoleh nilai 78,50, yang menandakan sistem cukup efisien, meskipun beberapa responden mencatat bahwa waktu loading masih dapat dipercepat untuk meningkatkan kelancaran interaksi. Aspek memorability memperoleh nilai 81,25, yang menunjukkan bahwa pengguna dapat dengan mudah mengingat kembali cara menggunakan sistem setelah jeda waktu tertentu, berkat konsistensi visual dan tata letak yang logis. Aspek error handling menunjukkan nilai tertinggi sebesar 87,75, mengindikasikan bahwa sistem berjalan stabil tanpa gangguan signifikan, dengan hanya beberapa kendala minor seperti sensitivitas gerak tangan yang bervariasi antarperangkat. Sementara itu,

aspek satisfaction memperoleh nilai 85,50, yang menegaskan bahwa pengguna merasa puas secara emosional terhadap pengalaman yang diberikan, baik dari segi visualisasi, performa, maupun relevansi konten edukatifnya.



Gambar 9. 2 Perbandingan Nilai SUS antar Kelompok Responden

Gambar 9.2 menunjukkan perbandingan hasil antar kelompok menunjukkan variasi persepsi yang menarik. Kelompok Bappeda memperoleh rata-rata 79,16, dengan fokus utama pada aspek efficiency dan accuracy, di mana mereka menilai sistem potensial untuk mendukung perencanaan berbasis spasial namun masih memerlukan peningkatan pada kecepatan akses data. Kelompok ahli game mencatat skor tertinggi, yaitu 84,21, dengan penilaian positif pada stabilitas sistem dan kehalusan interaction design, terutama dalam hal respons gerak dan rendering. Sementara itu, kelompok mahasiswa memperoleh rata-rata 82,00, dengan penekanan pada aspek learnability dan satisfaction, menilai sistem mudah digunakan, interaktif, dan menyenangkan sebagai media pembelajaran tentang

lingkungan perkotaan. Temuan ini menunjukkan bahwa persepsi *usability* sangat dipengaruhi oleh latar belakang pengalaman pengguna profesional menilai dari sisi fungsionalitas dan efisiensi, sedangkan pengguna akademik dan teknolog lebih menekankan pada kenyamanan dan pengalaman imersif.

Secara keseluruhan, hasil pengujian menunjukkan bahwa seluruh dimensi *usability* telah terpenuhi dengan baik, dengan rata-rata skor antar aspek yang relatif seimbang: *learnability* (82,00), *efficiency* (78,50), *memorability* (81,25), *error handling* (87,75), dan *satisfaction* (85,50). Skor rata-rata 81,80 menempatkan Monitoring Malang City jauh di atas nilai benchmark global SUS (68), menegaskan bahwa sistem ini telah mencapai standar profesional dalam hal kemudahan penggunaan dan kepuasan pengguna. Meskipun terdapat ruang peningkatan pada sisi efisiensi dan optimisasi performa, hasil ini membuktikan keberhasilan pendekatan *user-centered design* yang diterapkan dalam pengembangan sistem. Dengan penyempurnaan teknis minor seperti optimisasi loading pipeline dan penambahan interactive tutorial, sistem ini berpotensi mencapai kategori Excellent ( $\geq 90$ ) pada iterasi pengujian berikutnya. Secara interpretatif, temuan ini menegaskan bahwa VR Monitoring Malang City bukan hanya berfungsi sebagai media edukasi interaktif, tetapi juga sebagai alat yang efektif untuk mendukung analisis spasial dan kesadaran lingkungan dalam konteks perencanaan kota yang berkelanjutan.

## BAB X

### PEMBAHASAN

Penelitian ini bertujuan untuk menganalisis performa metode SVR dalam memprediksi fenomena UHI di Kota Malang, serta mengimplementasikan hasil prediksi tersebut dalam bentuk visualisasi adaptif berbasis *Virtual Reality* (VR). Pendekatan ini mengintegrasikan data penginderaan jauh, pemodelan pembelajaran mesin, dan teknologi visualisasi spasial tiga dimensi untuk menghasilkan sistem yang tidak hanya akurat secara analitis, tetapi juga komunikatif dan interaktif dalam penyampaian informasi spasial. Berdasarkan hasil pengujian model, SVR dengan *kernel* Radial Basis Function (RBF) menunjukkan performa terbaik dengan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 2,76% dan koefisien determinasi ( $R^2$ ) sebesar 0,887. Nilai tersebut menandakan bahwa model mampu menjelaskan hampir 89% variasi data aktual terhadap hasil prediksi. Kinerja ini memperlihatkan kemampuan SVR dalam menangani hubungan non-linear antara variabel lingkungan, seperti *Normalized Difference Vegetation Index* (NDVI), *Normalized Difference Built-up Index* (NDBI), *Normalized Difference Moisture Index* (NDMI), *Urban Density Index* (UDI), elevasi, serta suhu permukaan tanah (LST). Hasil ini juga menunjukkan bahwa *kernel* RBF memiliki kemampuan generalisasi yang lebih baik dibandingkan *kernel* Linear maupun Polynomial, sehingga dinilai paling sesuai untuk memodelkan fenomena lingkungan yang kompleks.

Integrasi antara model SVR dan sistem visualisasi dilakukan melalui *Application Programming Interface* (API) berbasis *FastAPI* yang berfungsi sebagai penghubung antara *backend* pemrosesan data dan *frontend* visualisasi di *Unity 3D*. Sistem ini dirancang dengan arsitektur dua lapis (*two-tier architecture*) yang terdiri atas *Data Processing Layer* dan *Visualization Layer*. Melalui mekanisme komunikasi asinkron, sistem mampu menampilkan hasil prediksi suhu permukaan tanah secara *real-time* dengan presisi spasial mencapai 0,3 meter. Proses integrasi ini memungkinkan pembaruan visualisasi data suhu tanpa perlu pemrosesan ulang secara manual, sekaligus meningkatkan efisiensi sistem.

Dari sisi visualisasi, implementasi dalam *Unity 3D* memanfaatkan *Finite State Machine* (FSM) sebagai pengendali logika interaksi pengguna. FSM mengatur transisi antar *state* seperti *Start*, *Tutorial*, *Exploration*, *Monitoring*, dan *Completion* agar permainan berlangsung sistematis dan terarah. Pengguna dapat berinteraksi dengan lingkungan kota virtual, menjelajahi titik-titik panas (*hotspot*), dan menganalisis hubungan antara variabel lingkungan dan peningkatan suhu secara interaktif. Desain antarmuka yang adaptif memudahkan pengguna dalam memahami data, sementara sistem pewarnaan termal (*heatmap*) memberikan visualisasi yang intuitif terhadap distribusi suhu di Kota Malang.

Evaluasi *usability* menggunakan metode *System Usability Scale* (SUS) menunjukkan skor rata-rata di atas 80, yang dikategorikan sebagai “**Excellent**”. Artinya, sistem visualisasi VR yang dikembangkan memiliki tingkat kemudahan penggunaan yang tinggi, tampilan yang menarik, serta efektivitas yang baik dalam menyampaikan informasi. Lima aspek *usability* menurut Nielsen yakni

*Learnability, Efficiency, Memorability, Errors, dan Satisfaction* menunjukkan hasil positif, terutama pada aspek *Efficiency* dan *Satisfaction*, yang menandakan sistem ini efisien digunakan dan memberikan pengalaman interaktif yang menyenangkan. Dengan demikian, hasil penelitian ini menegaskan bahwa penggabungan SVR, API, dan VR dapat menghasilkan sistem visualisasi adaptif yang tidak hanya memberikan akurasi prediksi tinggi, tetapi juga meningkatkan pemahaman fenomena termal perkotaan secara signifikan. Pendekatan ini mengubah data ilmiah yang bersifat abstrak menjadi representasi spasial yang mudah dipahami, sekaligus memberikan kontribusi terhadap pengembangan teknologi *smart city* dan pengelolaan lingkungan perkotaan berbasis data ilmiah.

Dengan mendasarkan penelitian ini pada prinsip-prinsip agama melalui integrasi islam yaitu berdasarkan Al-Quran dan hadist, diharapkan mampu menekankan pentingnya perlindungan lingkungan dan kesadaran akan peran manusia dalam menjaga bumi sebagai amanah dari Allah. Dalam mengamalkan wujud dari muamalah mu'allah yaitu dengan bertaqwa kepada Allah dan memperhatikan apa yang telah diperbuat serta memperhatikan hari esok merupakan wujud dari muamalah mu'allah seperti tertera pada surat Al-Hasyr : 18 yang berbunyi.

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَلْتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِغَدٍ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ خَبِيرٌ بِمَا تَعْمَلُونَ ۝١

*“Wahai orang-orang yang beriman! Bertakwalah kepada Allah dan hendaklah setiap orang memperhatikan apa yang telah diperbuatnya untuk hari esok (akhirat), dan bertakwalah kepada Allah. Sungguh, Allah Mahateliti terhadap apa yang kamu kerjakan.” (Q.S. Al-Hasyr : 18).*

Menurut tafsir al madinah al-munawwarah, Allah memerintahkan hamba-hamba-Nya yang beriman untuk mentaati segala perintah-Nya dan menjaUHI segala larangan-Nya, dan hendaklah setiap mereka mencermati kebaikan apa yang telah dia siapkan untuk menghadapi hari kiamat. Kemudian Allah mengulangi perintah-Nya agar mereka bertakwa kepada Allah, untuk menegaskan betapa pentingnya ketakwaan. Allah Maha Mengetahui segala perbuatan kalian di dunia.

Dari tafsir diatas dapat dijelaskan pentingnya introspeksi diri dan muhasabah. Setiap manusia diharapkan untuk selalu mengkaji dirinya sendiri dengan selalu mentaati perintahNya dan jika terdapat kesalahan sebelumnya, hendaknya segera berusaha untuk memperbaikinya dengan melepaskan diri dari dosa, bertaubat dengan sungguh-sungguh, dan menjauh dari segala hal yang dapat mengantarkan pada kesalahan sebelumnya. Dalam konteks prediksi fenomena UHI ini mengajarkan untuk selalu bertaqwa kepada Allah dengan memperhatikan segala dampak dari perbuatan manusia untuk kedepannnya baik didunia dan di akhirat kelak. Dengan menjaga Amanah Allah yang sudah diberikan, berupa lingkungan yang indah serta terjaga. Ini merupakan bentuk ketaqwaan kepada Allah. Dengan mempertimbangkan dampak aktivitas dan perbuatan manusia untuk perlindungan lingkungan di masa depan, yang menjadi tanggung jawab manusia sebagai al aziz. Dengan adanya prediksi fenomena UHI diharapkan mampu menjadi jembatan untuk perencanaan tata Kelola kota yang baik, dengan mempertimbangkan aspek aspek yang berpengaruh terhadap lingkungan kota khususnya untuk mengatasi adanya fenomena UHI.

Diharapkan penelitian ini dapat membantu pemerintahan kota serta pihak lainnya untuk saling membantu mengatasi adanya fenomena UHI di kota Malang. Dalam QS Al-Maidah pada ayat 2 dijelaskan bahwa: .

...وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ ۖ

”..Dan tolong-menolonglah kamu dalam berbuat kebaikan dan perbuatan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran.”(QS. Al-Maidah : 2).

Ibnu Katsir menjelaskan Allah memerintahkan seluruh hambaNya yang mengaku beriman untuk saling tolong-menolong dalam melakukan hal kebaikan dan menjauhi setiap larangannya yang berujung pada perbuatan dosa. Allah mengharamkan manusia untuk saling tolong-menolong dalam hal buruk yang dapat mengakibatkan kerusakan, keburukan, maksiat, dan perilaku haram lainnya.

Salah satu contoh contoh yang baik yang telah disabdakan oleh Rasulullah:

انصُرْ أَخَاكَ ظَالِمًا أَوْ مَظْلُومًا قَالُوا يَا رَسُولَ اللَّهِ هَذَا نَنْصُرُهُ مَظْلُومًا فَكَيْفَ نَنْصُرُهُ ظَالِمًا قَالَ تَأْخُذُ فَوْقَ يَدَيْهِ

“Bantulah saudaramu, baik dalam keadaan sedang berbuat zhalim atau sedang teraniaya. Ada yang bertanya: “Wahai Rasulullah, kami akan menolong orang yang teraniaya. Bagaimana menolong orang yang sedang berbuat zhalim?” Beliau menjawab: “Dengan menghalanginya melakukan kezhaliman. Itulah bentuk bantuanmu kepadanya.” [HR. al-Bukhâri]

Orang berilmu membantu orang lain dengan ilmunya. Orang kaya membantu dengan kekayaannya. Dan hendaknya kaum Muslimin menjadi satu tangan dalam membantu orang yang membutuhkan. Oleh karena itu, perbuatan tolong-menolong dalam kebaikan diharapkan mampu menjadikan manusia menjadi masnusia yang bermanfaat bagi manusia lainnya, dan sebaik-baiknya



manusia adalah manusia yang bermanfaat bagi sekitarnya seperti pada. Hadis yang telah diriwayatkan oleh Imam Ahmad:

*“Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia.”(HR. Ahmad).*

Syeikh Ali bin Hasan Al-Halabi dalam kitab Ahkamusy Syita’ Fis Sunnatil Muthahharah menjelaskan bahwa Melakukan dan mempercayai prediksi suatu fenomena itu diperbolehkan, akan tetapi dengan syarat kepercayaan hanya untuk melakukan evaluasi untuk mempersiapkan fenomena yang akan datang, akan tetapi tidak boleh percaya dengan utuh serta mengabarkan bahwa peristiwa yang telah diprediksikan sebelumnya pasti terjadi

Menurut tafsir tersebut, mempercayai prediksi fenomena seperti UH) diperbolehkan sebagai langkah untuk mempersiapkan diri dan melakukan evaluasi terhadap dampak lingkungan kedepannya. Namun, penting untuk tidak menganggap hasil prediksi tersebut sebagai sesuatu yang pasti akan terjadi di masa depan. Hal ini karena meyakini secara mutlak atas sesuatu yang belum terjadi merupakan tindakan yang mendahului kuasa Allah dan termasuk pada perbuatan syirik. Dalam Islam, keyakinan terhadap takdir Allah merupakan prinsip fundamental. Meskipun kita dapat merencanakan dan mengantisipasi berbagai fenomena berdasarkan ilmu pengetahuan dan penelitian, akhirnya keputusan mutlak tetaplah di tangan Allah. Oleh karena itu, penting untuk berusaha menjaga keseimbangan antara usaha manusia khususnya dalam mengatasi fenomena UHI serta kekuasaan dan takdir Allah.

Fenomena kerusakan yang terjadi di bumi ini tidak hanya disebabkan oleh faktor alam dan takdir Allah saja, namun juga faktor yang disebabkan oleh manusia itu sendiri sangat berpengaruh, seperti contohnya peristiwa banjir yang disebabkan oleh saluran air yang terhambat dikarenakan adanya sampah yang dibuang sembarangan dan dihasilkan oleh manusia itu sendiri serta akibat lainnya dari penebangan serta pembakaran hutan yang dapat mengurangi daerah resapan air oleh tanah. Kemudian peristiwa tanah longsor yang diakibatkan oleh faktor paling dominan dari penambangan ataupun pengikisan batu bukit oleh manusia. contoh lainnya pada fenomena UHI yang diakibatkan oleh perukaan lahan alami menjadi lahan yang terbangun dan kedap air, sehingga mengakibatkan ketidaksimbangan alam yang ditunjukkan dengan kenaikan suhu di permukaan yang menjadi tanda bahwa adanya fenomena UHI. Berikut adalah salah satu ayat Qur'an yang berhubungan yaitu Q.S Ar-Rum:41-42 :

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ ٤١ قُلْ سِيرُوا فِي الْأَرْضِ فَانظُرُوا كَيْفَ كَانَ عَاقِبَةُ الَّذِينَ مِن قَبْلُ ۚ كَانَ أَكْثَرُهُم مُّشْرِكِينَ ٤٢

*“Telah tampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebahagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar). Katakanlah: “Adakanlah perjalanan di muka bumi dan perhatikanlah bagaimana kesudahan orang-orang yang terdahulu. kebanyakan dari mereka itu adalah orang-orang yang mempersekutukan (Allah).” (QS. Ar Rum : 41-42).*

Dalam tafsir Ibnu Katsir dijelaskan bahwa ayat ini mengingatkan manusia agar tidak berbuat kerusakan di muka bumi setelah Allah memberikan nikmat dan rezeki kepada mereka. Allah memerintahkan agar manusia menggunakan nikmat tersebut dengan baik dan tidak menyalahgunakannya sehingga menimbulkan

kerusakan dan kerusakan yang dilakukan serta diakibatkan dari manusia sendiri telah terjadi sejak dahulu, merupakan bagian dari peringatan dari Allah agar manusia selalu ingat akan akibat dari perbuatannya dan selalu berusaha untuk menjaga keseimbangan dan kelestarian lingkungan..

Dalam ayat lain, yaitu pada QS. Al-Baqarah : 60 :

﴿وَإِذِ اسْتَسْقَىٰ مُوسَىٰ لِقَوْمِهِ فَقُلْنَا اضْرِبْ بِعَصَاكَ الْحَجَرَ فَانْفَجَرَتْ مِنْهُ اثْنَتَا عَشْرَةَ عَيْنًا ۖ قَدْ عَلِمَ كُلُّ أُنَاسٍ مَّشْرَبَهُمْ ۖ كُلُوا وَاشْرَبُوا مِنْ رِّزْقِ اللَّهِ وَلَا تَعْنُوا فِي الْأَرْضِ مُفْسِدِينَ ۚ ۖ﴾

*“Dan (ingatlah) ketika Musa memohon air untuk kaumnya, lalu Kami berfirman, “Pukullah batu itu dengan tongkatmu!” Maka memancarlah daripadanya dua belas mata air. Setiap suku telah mengetahui tempat minumnya (masing-masing). Makan dan minumlah dari rezeki (yang diberikan) Allah, dan janganlah kamu melakukan kejahatan di bumi dengan berbuat kerusakan.”(QS. Al-Baqarah : 60).*

Dalam tafsir yang diriwayatkan oleh Ibnu Katsir bahwa ayat ini menceritakan kisah Bani Israil yang tidak bersyukur atas nikmat yang Allah berikan kepada mereka. Mereka bahkan melakukan kerusakan dan memilih cara yang mereka anggap lebih baik daripada apa yang telah Allah tetapkan. Ini menunjukkan bahwa manusia sering tidak menghargai nikmat Allah dan malah menyebabkan kerusakan di bumi..

Berdasarkan tafsir diatas, yang dicontohkan melalui bani israil yang melakukan kerusakan. Jika dibandingkan pada zaman ini, manusia juga melakukan kerusakan alam. Banyak sekali contoh nyata yang sekarang mulai dirasakan oleh manusia sendiri. Diantaranya, Lahan vegetasi seringkali dikonversi menjadi lahan yang terbangun, ditandai dengan pembangunan bangunan tinggi di perkotaan dan penggunaan fasilitas yang berdampak negatif seperti penggunaan bahan kimia berbahaya dan polusi udara. Perubahan ini mengakibatkan hilangnya

keseimbangan ekosistem serta meningkatkan suhu kota yang menjadi tanda terjadinya fenomena UHI. Manusia perlu mengambil pelajaran dari ayat tersebut untuk bertanggung jawab dalam memelihara bumi sesuai amanah Allah dan menghindari tindakan yang merusak lingkungan demi kepentingan sesaat.

## BAB XI

### PENUTUP

#### 11.1 Kesimpulan

Berdasarkan hasil penelitian, analisis, dan implementasi yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

- a. Model SVR dengan *kernel* RBF merupakan model paling optimal dalam memprediksi suhu permukaan tanah (LST) di Kota Malang, dengan nilai MAPE sebesar 2,76% dan  $R^2$  sebesar 0,887 yang menunjukkan tingkat akurasi dan kemampuan generalisasi model yang baik terhadap data lingkungan perkotaan. Hal ini membuktikan bahwa faktor-faktor penentu seperti kepadatan bangunan (UDI), vegetasi (NDVI), permukaan kedap air (NDBI), kelembapan (NDMI), air (NDWI), serta elevasi dari data SRTM memiliki hubungan yang kuat terhadap LST.
- b. Implementasi visualisasi adaptif berbasis VR berhasil menampilkan fenomena UHI secara imersif, interaktif, dan edukatif, sehingga memudahkan pengguna memahami distribusi suhu perkotaan dan pengaruh variabel lingkungan secara spasial. Evaluasi *usability* menggunakan SUS menunjukkan skor di atas rata-rata ( $\geq 68$ ), yang menandakan sistem mudah digunakan, efisien, dan memberikan pengalaman positif bagi pengguna.

Secara keseluruhan, sistem visualisasi UHI berbasis SVR, API, dan VR yang dikembangkan dalam penelitian ini berhasil menciptakan simulasi lingkungan yang realistis, akurat, dan bermanfaat untuk mendukung analisis, edukasi, serta pengambilan keputusan dalam perencanaan kota berkelanjutan.

## 11.2 Saran

Berdasarkan temuan dan keterbatasan dalam penelitian ini, beberapa saran untuk pengembangan dan penelitian selanjutnya adalah sebagai berikut:

- a. Penelitian berikutnya disarankan untuk menambahkan data multi-temporal dan multi-sumber, seperti citra MODIS atau ERA5-Land, guna meningkatkan resolusi temporal dan akurasi prediksi suhu permukaan.
- b. Pengembangan sistem ke arah visualisasi dinamis berbasis waktu (day-night cycle) akan memberikan representasi yang lebih realistis terhadap variasi termal harian di kawasan perkotaan.
- c. Evaluasi *usability* dapat diperluas dengan menggunakan instrumen tambahan seperti NASA-TLX, UEQ, atau PSSUQ untuk menilai beban kerja mental dan aspek emosional pengguna secara lebih mendalam.
- d. Implementasi sistem multi-user dalam VR dapat dikembangkan menjadi lingkungan kolaboratif, di mana beberapa pengguna dapat berinteraksi secara simultan untuk mendukung pembelajaran dan perencanaan kota partisipatif.
- e. Sistem ini memiliki potensi besar untuk dikembangkan sebagai Decision Support System (DSS) bagi pemerintah daerah, akademisi, dan perencana kota dalam memantau, menganalisis, serta merumuskan kebijakan mitigasi fenomena UHI secara berkelanjutan.

## DAFTAR PUSTAKA

- Addas, A. (2023). Machine Learning Techniques to Map the Impact of Urban Heat Island: Investigating the City of Jeddah. *Land*, 12(6). <https://doi.org/10.3390/land12061159>
- Bokaie, M., Zarkesh, M. K., Arasteh, P. D., & Hosseini, A. (2016). Assessment of Urban Heat Island based on the relationship between land surface temperature and Land Use/ Land Cover in Tehran. *Sustainable Cities and Society*, 23, 94–104. <https://doi.org/10.1016/j.scs.2016.03.009>
- Dede, M., Widiawaty, M. A., Nurhanifah, N., Ismail, A., Artati, A. R. P., Ati, A., & Ramadhan, Y. R. (2020). Estimasi Perubahan Kualitas Udara Berbasis Citra Satelit Penginderaan Jauh Di Sekitar PLTU Cirebon. *Jambura Geoscience Review*, 2(2), 78–87. <https://doi.org/10.34312/jgeosrev.v2i2.5951>
- Evy Sulistianingsih, S. S. M. (2018). Prediksi Nilai Tukar Dolar Amerika Serikat Terhadap Rupiah Dengan Metode Support Vector Regression (Svr). *Bimaster : Buletin Ilmiah Matematika, Statistika Dan Terapannya*, 8(1), 1–10. <https://doi.org/10.26418/bbimst.v8i1.30503>
- Fitria, R., Kim, D., Baik, J., & Choi, M. (2019). Impact of Biophysical Mechanisms on Urban Heat Island Associated with Climate Variation and Urban Morphology. *Scientific Reports*, 9(1), 1–13. <https://doi.org/10.1038/s41598-019-55847-8>
- Furuya, M. T. G., Furuya, D. E. G., de Oliveira, L. Y. D., da Silva, P. A., Cicerelli, R. E., Gonçalves, W. N., Junior, J. M., Osco, L. P., & Ramos, A. P. M. (2023). A machine learning approach for mapping surface urban heat island using environmental and socioeconomic variables: a case study in a medium-sized Brazilian city. *Environmental Earth Sciences*, 82(13), 1–14. <https://doi.org/10.1007/s12665-023-11017-8>
- Kamali Maskooni, E., Hashemi, H., Berndtsson, R., Daneshkar Arasteh, P., & Kazemi, M. (2021). Impact of spatiotemporal land-use and land-cover changes on surface urban heat islands in a semiarid region using Landsat data. *International Journal of Digital Earth*, 14(2), 250–270. <https://doi.org/10.1080/17538947.2020.1813210>
- Kusumadewi, T., Maryunani, Surjono, Arif, Y. M., Samudro, H., Samudro, G., & Mangkoedihardjo, S. (2023). Urban Phytoarchitecture Design Options: Greenspace Orientation and Tree Species Intensification. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 31(1), 183–196. <https://doi.org/10.37934/araset.31.1.183196>
- Li, F., Yigitcanlar, T., Nepal, M., Nguyen, K., & Dur, F. (2023). Machine learning

and remote sensing integration for leveraging urban sustainability: A review and framework. *Sustainable Cities and Society*, 96(March), 104653. <https://doi.org/10.1016/j.scs.2023.104653>

Li, J., Sun, R., Liu, T., Xie, W., & Chen, L. (2021). Prediction models of urban heat island based on landscape patterns and anthropogenic heat dynamics. *Landscape Ecology*, 36(6), 1801–1815. <https://doi.org/10.1007/s10980-021-01246-2>

Liu, S., Zhang, J., Li, J., Li, Y., Zhang, J., & Wu, X. (2021). Simulating and mitigating extreme urban heat island effects in a factory area based on machine learning. *Building and Environment*, 202(June), 108051. <https://doi.org/10.1016/j.buildenv.2021.108051>

*Malang Dalam Angka*. (2024).

Mathew, A., Khandelwal, S., & Kaul, N. (2016). Spatial and temporal variations of urban heat island effect and the effect of percentage impervious surface area and elevation on land surface temperature: Study of Chandigarh city, India. *Sustainable Cities and Society*, 26(1), 264–277. <https://doi.org/10.1016/j.scs.2016.06.018>

Mathew, A., Sreekumar, S., Khandelwal, S., & Kumar, R. (2019). Prediction of land surface temperatures for surface urban heat island assessment over Chandigarh city using support vector regression model. *Solar Energy*, 186(6), 404–415. <https://doi.org/10.1016/j.solener.2019.04.001>

Nainggolan, Y. C., Sasmito, B., Sukmono, A., Amaliana, D. R. Y. P. A. S., Muzaky, H., Jaelani, L. M., Ardi, I. R., Lubis, M. S., Fitrianiingsih, Y., Setioningrum, D., Risdiyanto, I., Kasus, S., Semarang, K., García Reyes, L. E., Syamsudin, F., Lestari, S., Novianto, A., Syahtriawan, Q. Z., Kusumawardani, D., ... Liu, Q. (2020). Analisis Kontributor Dominan Terhadap Fenomena Urban Heat Island di Kota Medan. *Skripsi: Institut Pertanian Bogor. Bogor (ID)*, 3(1), 1689–1699. [https://repository.ipb.ac.id/handle/123456789/40861%0Ahttp://repository.ipb.ac.id/handle/123456789/48333%0Ahttp://repository.ipb.ac.id/bitstream/123456789/11005/1/Nur Ikhwan Khusaini\\_E2008\\_abstract.pdf%0Ahttps://repository.ipb.ac.id/handle/123456789/62420](https://repository.ipb.ac.id/handle/123456789/40861%0Ahttp://repository.ipb.ac.id/handle/123456789/48333%0Ahttp://repository.ipb.ac.id/bitstream/123456789/11005/1/Nur%20Ikhwan%20Khusaini_E2008_abstract.pdf%0Ahttps://repository.ipb.ac.id/handle/123456789/62420)

Odindi, J. O., Bangamwabo, V., & Mutanga, O. (2015). Assessing the value of urban green spaces in mitigating multi-seasonal urban heat using MODIS land surface temperature (LST) and landsat 8 data. *International Journal of Environmental Research*, 9(1), 9–18.

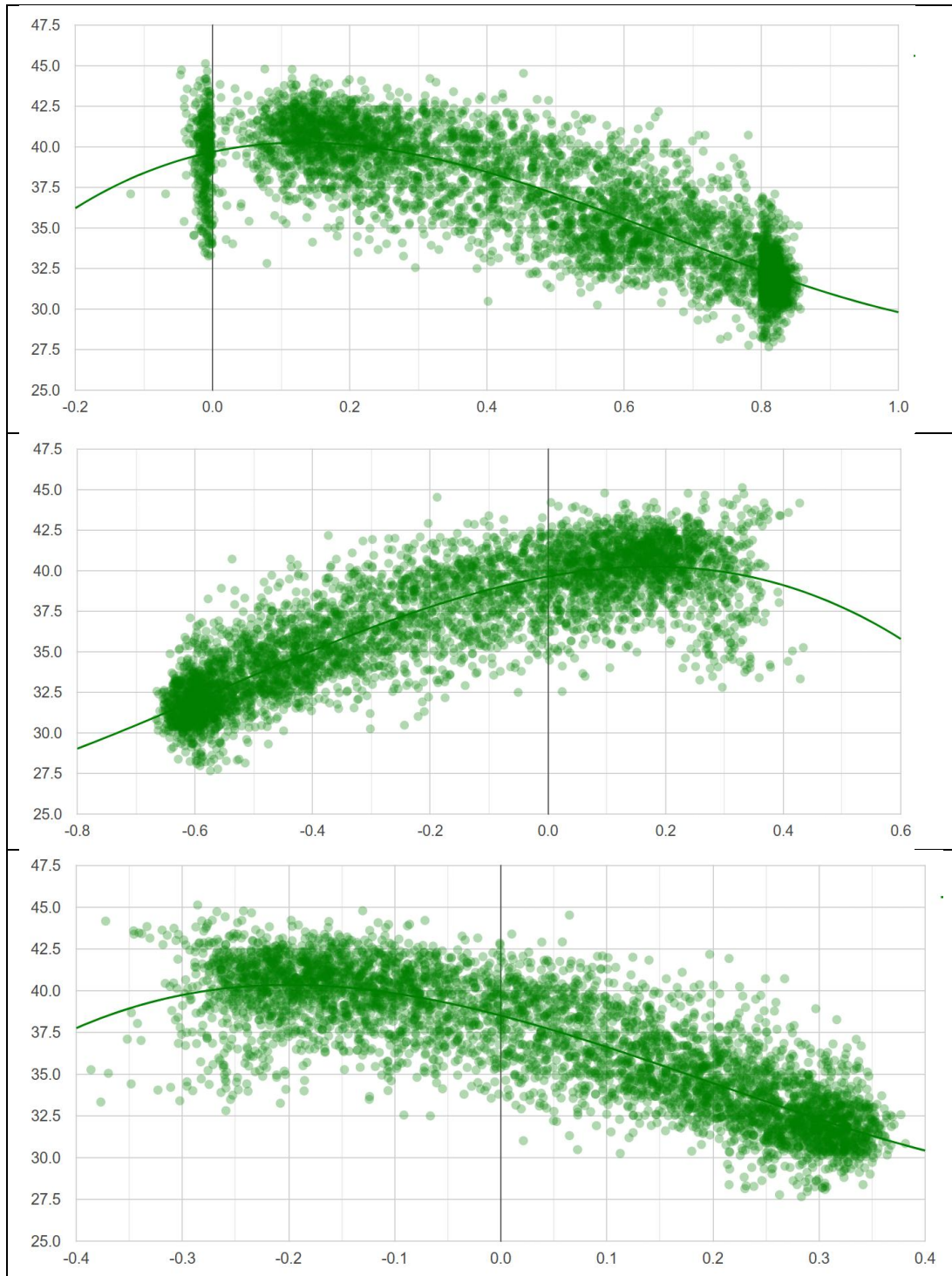
Peng, J., Dan, Y., Qiao, R., Liu, Y., Dong, J., & Wu, J. (2021). How to quantify the cooling effect of urban parks? Linking maximum and accumulation perspectives. *Remote Sensing of Environment*, 252. <https://doi.org/10.1016/j.rse.2020.112135>

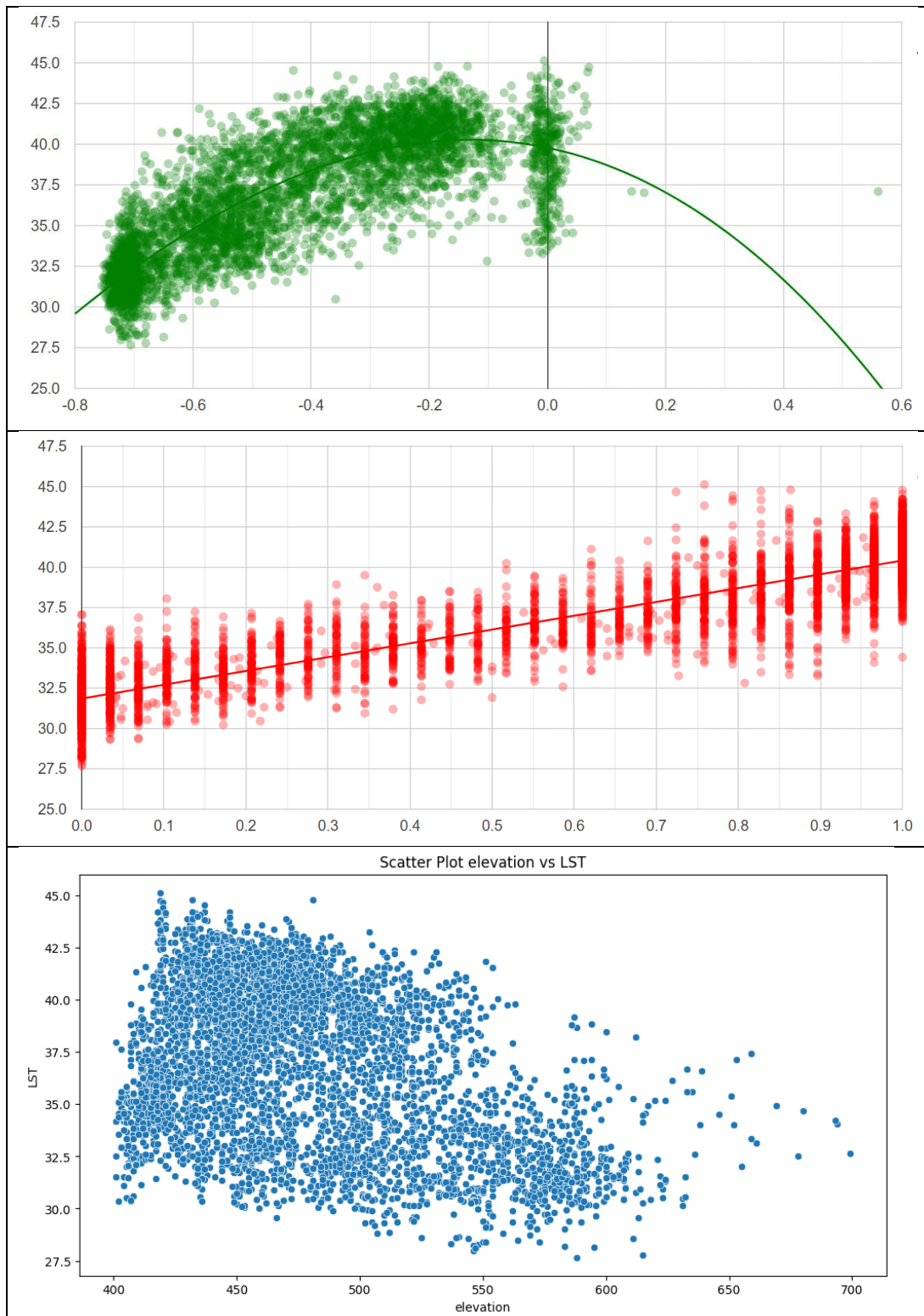


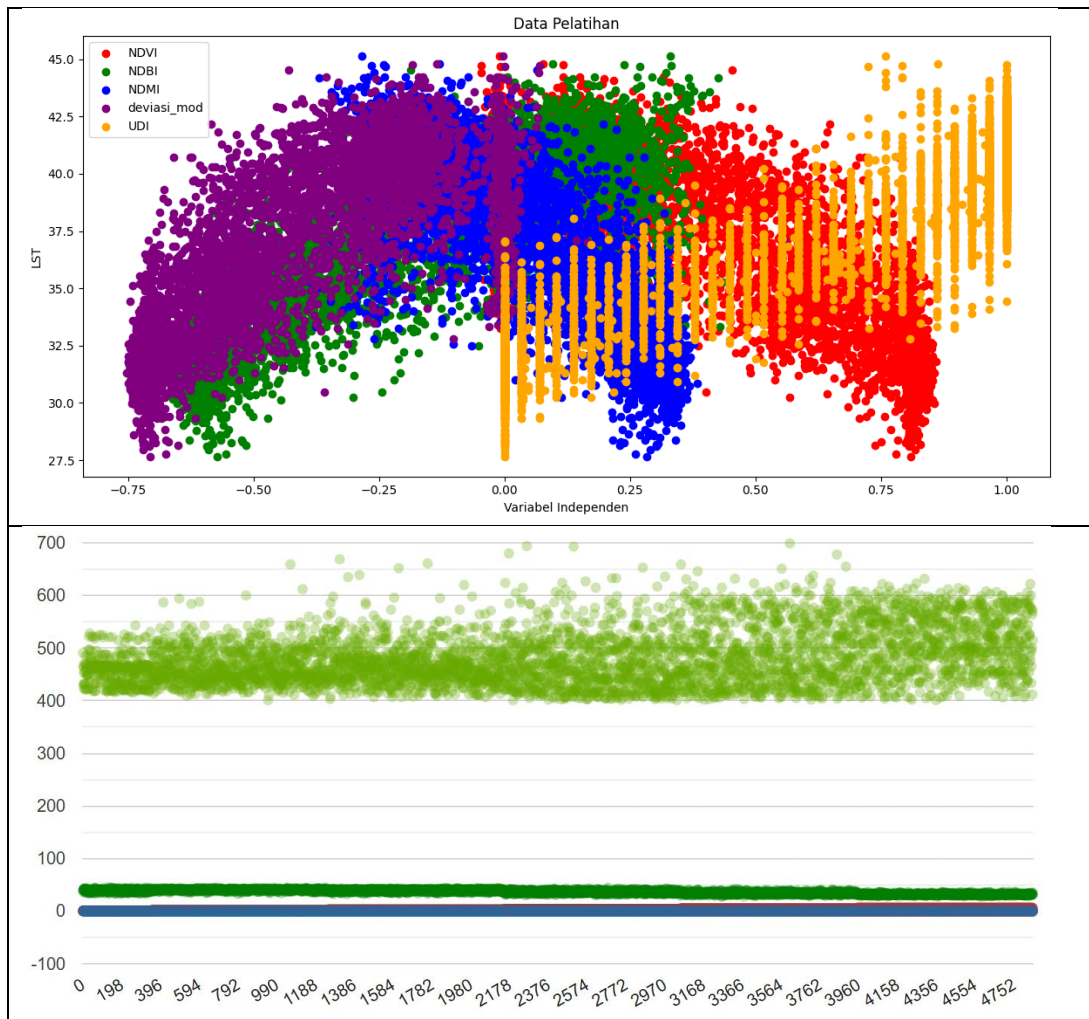
- Ren, T., Zhou, W., & Wang, J. (2021). Beyond intensity of urban heat island effect: A continental scale analysis on land surface temperature in major Chinese cities. *Science of the Total Environment*, 791, 148334. <https://doi.org/10.1016/j.scitotenv.2021.148334>
- Sannigrahi, S., Bhatt, S., Rahmat, S., Uniyal, B., Banerjee, S., Chakraborti, S., Jha, S., Lahiri, S., Santra, K., & Bhatt, A. (2018). Analyzing the role of biophysical compositions in minimizing urban land surface temperature and urban heating. *Urban Climate*, 24(September), 803–819. <https://doi.org/10.1016/j.uclim.2017.10.002>
- Santamouris, M. (2020). Recent progress on urban overheating and heat island research. Integrated assessment of the energy, environmental, vulnerability and health impact. Synergies with the global climate change. In *Energy and Buildings* (Vol. 207). <https://doi.org/10.1016/j.enbuild.2019.109482>
- Shi, H., Xian, G., Auch, R., Gallo, K., & Zhou, Q. (2021). Urban heat island and its regional impacts using remotely sensed thermal data—a review of recent developments and methodology. *Land*, 10(8). <https://doi.org/10.3390/land10080867>
- Sun, Y., Gao, C., Li, J., Wang, R., & Liu, J. (2019). Quantifying the Effects of Urban Form on Land Surface Temperature in Subtropical High-Density Urban Areas Using Machine Learning. *Remote Sensing*, 11(8), 959. <https://doi.org/10.3390/rs11080959>
- Zhang, F., & O'Donnell, L. J. (2019). Support vector regression. In *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-815739-8.00007-9>
- Zhang, X., Estoque, R. C., & Murayama, Y. (2017). An urban heat island study in Nanchang City, China based on land surface temperature and social-ecological variables. *Sustainable Cities and Society*, 32, 557–568. <https://doi.org/10.1016/j.scs.2017.05.005>

# LAMPIRAN

## Lampiran 1 Data Non-Linier







## Lampiran 2 Psudocode unity web request ke server

```

FUNCTION PredictFromCsv(optional dateISO):
    # Kirim permintaan prediksi CSV ke server
    SET url = baseUrl + "/gee/predict-from-csv"
    # Tentukan endpoint API
    IF dateISO IS EMPTY:
        SET dateISO = current UTC date in "yyyy-MM-dd" format
        # Gunakan tanggal hari ini jika kosong
    IF fixedDateISO IS NOT EMPTY:
        SET dateISO = fixedDateISO
        # Override jika ada tanggal tetap
    DISPLAY " POST (CSV): " + url ON statusText
    # Tampilkan status request
    CALL StartProgress("CSV")
    # Mulai animasi loading
    CREATE requestBody AS JSON OBJECT:
        {
            date = dateISO,
            window_days = windowDays,
            backfill_days = backfillDays,
            model_feature_order = featureOrder,
            save_csv_path = null
        }
    # Siapkan body JSON
    CONVERT requestBody TO JSON STRING
    # Ubah ke format string JSON
    INITIALIZE UnityWebRequest (POST, url)
    SET timeout = csvTimeoutSeconds
    SET uploadHandler = JSON string as UTF8 bytes
    SET downloadHandler = buffer
    SET headers:
        "Content-Type" = "application/json"
        "Accept" = "application/json"
    # Atur konfigurasi request dan header
    SEND request TO server
    # Kirim ke server
    WAIT for response
    # Tunggu hasil
    CALL StopProgress()
    # Hentikan loading
    IF response IS SUCCESSFUL:
        PARSE response JSON INTO object "resp"
        # Parse hasil ke objek
        IF resp IS NULL OR resp.results IS EMPTY:
            DISPLAY " Resp kosong. Mungkin window_days terlalu sempit." ON statusText
            TRIGGER OnGeeFailed("Empty response")
            # Tangani jika hasil kosong
        ELSE:
            CALL SaveResponseToCsv(resp) → savedFilePath
            # Simpan hasil ke CSV
            GET firstResult = resp.results[0]
            # Ambil data pertama
            DISPLAY SUCCESS MESSAGE:
                " {resp.n_points} titik | tgl={resp.date_used} |
                LST0={firstResult.lst_denorm}°C\nSaved: {savedFilePath}"
            # Tampilkan info hasil
            IF savedFilePath IS NOT EMPTY:
                TRIGGER OnGeeCompleted(savedFilePath)
                # Panggil event sukses
        ELSE:
            SET code = responseCode
            SET body = responseText
            DISPLAY "CSV predict gagal [code] result | error\nbody" ON statusText
            TRIGGER OnGeeFailed("HTTP {code}: {error}")
            # Tangani error HTTP
    END FUNCTION

```

### Lampiran 3 Psudocode Data Input, Validasi, dan Persiapan Data GEE

```

BEGIN
  INPUT: Request (req) dengan parameter csv_path, date,
  window_days, backfill_days, dan model_feature_order
  SET csv_path ← req.csv_path OR default_path #
  Tentukan jalur file CSV
  CONVERT csv_path ke absolute path #
  Pastikan jalur absolut
  IF file CSV tidak ditemukan THEN #
  Validasi keberadaan file
    THROW Error "CSV tidak ditemukan"
  END IF
  TRY
    READ CSV file menjadi dataframe df_pts #
  Membaca file CSV
  CATCH error
    THROW Error "Gagal membaca CSV"
  END TRY
  VALIDATE kolom wajib {type, longitude, latitude} #
  Pastikan kolom minimal ada
  IF kolom tidak lengkap THEN
    THROW Error "Kolom wajib hilang"
  END IF
  IF jumlah baris CSV == 0 THEN #
  Validasi CSV kosong
    THROW Error "CSV kosong"
  END IF
  IF jumlah baris CSV > 2500 THEN #
  Batasi jumlah titik
    THROW Error "Maks 2500 titik per permintaan"
  END IF
  INIT feats ← [] #
  List fitur Earth Engine
  FOR setiap baris i dalam df_pts DO
    lon, lat ← ambil nilai longitude dan latitude #
  Ambil koordinat
    BUAT ee.Feature dengan atribut pid, type, longitude,
    latitude
    ADD feature ke feats #
  Simpan fitur ke list
  END FOR
  SET pts_fc ← FeatureCollection(feats) #
  Buat FeatureCollection dari titik
  SET roi ← buffer geometry(pts_fc, 2000) #
  Area analisis dengan buffer 2 km
  SET date_iso ← req.date OR tanggal_hari_ini #
  Ambil tanggal analisis
  SET end ← ee.Date(date_iso) #
  Tanggal akhir
  SET start ← end.advance(-req.window_days, "day") #
  Tanggal mulai berdasarkan jendela hari
  END

```

## Lampiran 4 Psudocode Pemrosesan GEE, Prediksi Model, dan Output

```

BEGIN
  # --- Tahap Komposit dan Ekstraksi Citra dari GEE ---
  CALL sr_narrow ← modis_sr_composite(roi, start, end, backfill_days)
  # Komposit MODIS narrow window
  CALL sr_wide ← modis_sr_composite(roi, start-backfill, end+backfill)
  # Komposit MODIS wide window
  CALL sr_filled ← unmask(sr_narrow, sr_wide)
  # Isi celah citra yang hilang
  CALL lst ← modis_lst_c_filled(roi, start, end, backfill_days)
  # Ambil data LST
  CALL stack ← build_feature_stack(sr_filled, roi)
  # Gabungkan layer fitur ke stack
  CALL samples_fc ← sample_at_points(stack, lst, pts_fc, scale=500)
  # Sampling data di titik input
  TRY
    records ← samples_fc.getInfo()["features"]
  # Ambil hasil sampling
  CATCH error
    THROW Error "GEE sampling gagal"
  END TRY
  # --- Tahap Konversi dan Validasi Dataframe ---
  CONVERT records menjadi DataFrame df
  # Ubah hasil sampling ke tabel
  URUTKAN df berdasarkan pid dan reset index
  # Penataan data
  SET need ← req.model_feature_order
  # Urutan fitur model
  CHECK apakah fitur pada df lengkap terhadap need
  # Validasi fitur model
  IF ada fitur hilang THEN THROW Error "Fitur hilang"
  CHECK kecocokan jumlah fitur dengan model (n_model)
  # Validasi dimensi input model
  IF mismatch THEN THROW Error "Mismatch fitur"
  CHECK nilai NaN dalam df[need]
  # Validasi nilai kosong
  IF ditemukan THEN THROW Error "Nilai NaN ditemukan"
  # --- Tahap Prediksi Menggunakan Model ---
  SET X ← df[need]
  # Siapkan data input model
  TRY
    y ← svr.predict(X)
  # Jalankan prediksi
  CATCH error
    THROW Error "Gagal prediksi"
  END TRY
  TAMBAH kolom y_pred dan lst_denorm ke df
  # Simpan hasil prediksi dan denormalisasi
  # --- Tahap Penyimpanan dan Output ---
  IF req.save_csv_path diberikan THEN
  # Jika user ingin menyimpan hasil
    TENTUKAN out_path absolut
    SIMPAN df ke file CSV
  # Ekspor hasil prediksi
  END IF
  PILIH kolom output relevan (fitur, prediksi, lst_denorm)
  # Pilih kolom hasil
  KONVERSI setiap baris df ke objek PointOut
  # Siapkan hasil dalam format API
  RETURN GeePredictResp (tanggal, jumlah titik, fitur, hasil)
  # Kembalikan hasil prediksi

```

## Lampiran 5 Psudocode koordinat unity

```

CLASS PointPlacer
    VARIABLE csvFile // file CSV berisi data titik
    (tipe, longitude, latitude)
    VARIABLE defaultMarkerPrefab // prefab penanda default
    VARIABLE urbanMarkerPrefab // prefab untuk area urban
    (misal: warna merah)
    VARIABLE ruralMarkerPrefab // prefab untuk area rural
    (misal: warna hijau)
    VARIABLE terrain // referensi objek terrain di
    Unity
    VARIABLE autoPlaceOnValidate // opsi untuk menempatkan
    titik secara otomatis

    CONSTANT minLon, maxLon // batas longitude wilayah
    data
    CONSTANT minLat, maxLat // batas latitude wilayah
    data

    FUNCTION PlacePoints()
        // --- Hanya dijalankan di Unity Editor, bukan saat game
        dijalankan ---
        IF Application.isPlaying == TRUE THEN
            PRINT warning: "Harus dijalankan di Edit Mode"
            RETURN
        ENDIF

        // --- Hapus semua marker lama di dalam GameObject ini ---
        FOR setiap childObject DALAM transform
            IF childObject bukan aset (bukan prefab di Project
Folder)
                HAPUS segera (DestroyImmediate)
            ENDIF
        ENDFOR

        // --- Cek apakah CSV dan Terrain sudah diisi ---
        IF csvFile == NULL OR terrain == NULL THEN
            PRINT warning: "File CSV dan Terrain belum di-assign"
            RETURN
        ENDIF

        // --- Baca semua baris dari CSV ---
        lines = pisahkan teks csvFile berdasarkan '\n'

        FOR setiap baris dari index 1 sampai akhir (skip header)
            IF baris kosong THEN CONTINUE
            fields = pisahkan baris berdasarkan ','

            IF jumlah fields < 3 THEN CONTINUE

            // --- Ambil data tipe, longitude, latitude ---
            type = fields[0]
            lon = konversi fields[1] menjadi float
            lat = konversi fields[2] menjadi float

```



```

        // --- Tentukan prefab berdasarkan tipe lokasi ---
        prefabToUse = defaultMarkerPrefab
        IF type == "urban" THEN prefabToUse =
urbanMarkerPrefab
        ELSE IF type == "rural" THEN prefabToUse =
ruralMarkerPrefab
        ENDIF

        IF prefabToUse == NULL THEN
            PRINT warning: "Prefab tidak ditemukan untuk tipe
ini"
            CONTINUE
        ENDIF

        // --- Normalisasi longitude dan latitude ke rentang
[0,1] ---
        tX = (maxLon - lon) / (maxLon - minLon)
        tZ = (maxLat - lat) / (maxLat - minLat)

        // --- Konversi ke koordinat lokal terrain ---
        localX = tX * terrain.width
        localZ = tZ * terrain.length

        basePos = Vector3(
            terrain.position.x + localX,
            0,
            terrain.position.z + localZ
        )

        // --- Dapatkan tinggi permukaan terrain di posisi
tersebut ---
        height = terrain.SampleHeight(basePos) +
terrain.position.y

        // --- Tentukan posisi spawn marker ---
        spawnPos = Vector3(basePos.x, height + 0.05,
basePos.z)

        // --- Instansiasi marker di dunia Unity ---
        instance = buat prefabToUse di posisi spawnPos
        parent = transform (agar berada di dalam PointPlacer)

        // --- Tambahkan komponen MarkerInfo untuk menyimpan
metadata ---
        IF instance tidak punya MarkerInfo
            tambahkan komponen MarkerInfo
        ENDIF

        instance.longitude = lon
        instance.latitude = lat
        instance.isRural = (type == "rural")
    ENDFOR
END FUNCTION

```

```

        FUNCTION OnValidate()
            // Fungsi ini otomatis dipanggil ketika ada perubahan di
Inspector
            IF autoPlaceOnValidate == TRUE DAN game tidak sedang
dijalankan THEN
                // Jalankan PlacePoints() setelah proses serialization
selesai
                jalankan PlacePoints() dengan sedikit delay
            ENDIF
        END FUNCTION
    END CLASS

// =====
// Editor Customization
// =====
CLASS PointPlacerEditor
    EXTENDS UnityEditor.Editor

    FUNCTION OnInspectorGUI()
        // Tampilkan semua properti publik PointPlacer di
Inspector
        tampilkan default inspector

        // Tambahkan tombol manual untuk menjalankan PlacePoints()
        IF tombol "Place Points Now" ditekan THEN
            panggil PlacePoints() pada objek target
        ENDIF
    END FUNCTION
END CLASS

```

