

BAB II LANDASAN TEORI

2.1 Kajian Terhadap Penelitian yang terkait Sebelumnya

Berikut ini adalah kajian dari penelitian sebelumnya yangss menjadikan referensi dari melakukan penelitian ini: (Andini et al., 2018) dalam jurnalnya yang berjudul “*Analisis Sebaran Vegetasi Dengan Citra Satelit Sentinel Menggunakan Metode NDVI dan Segmentasi (Studi Kasus: Kabupaten Demak)*” dalam penelitian ini melakukan perbandingan antara metode NDVI dengan metode segmentasi, kabupaten Demak memiliki kerapatan vegetasi yang bervariasi. Hasil dari kedua metode diatas menunjukan sedikit perbedaan. Perhitungan menggunakan rumus sangat memperhatikan dalam wilayah sehingga terjadi perbedaan yang signifikan.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini : (Zauhairah et al., 2022) dalam jurnalnya yang berjudul “*Penentuan Pemetaan Kadar Air Tanah Optimal Pada Lahan Perkebunan Kelapa Sawit (STUDI KASUS: KEBUN CIKASUNGKA, PT PERKEBUNAN NUSANTARA VIII, CIMULANG, BOGOR)*” dalam penelitian ini melakukan perbandingan antara NDVI dengan NMDI dimana hasil penelitian pada NDVI ini berkisar antara 0,09 dan 0,76 sementara itu hasil penelitian NMDI berkisaran pada 0,42 sampai 0,91.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini : (Mabrur, 2021) dalam jurnalnya yang berjudul “*Analisa Perbandingan Object Counting Dengan Ecognition Dan Picterra*” dalam penelitian ini hasil penelitian menunjukan bahwa penggunaan metode objek secara otomatis meningkatkan efisiensi waktu dan proses pengolahan lebih mudah, tetapi

tidak menghilangkan kemungkinan bahwa hasil perhitungan objek mungkin mengandung kesalahan yang harus diidentifikasi.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini : (Simarmata et al., 2021) dalam jurnal yang berjudul *“Analisis Transformasi Index NDVI, NDWI Dan SAVI Untuk Identifikasi Kerapatan Vegetasi Mangrove Menggunakan Citra Sentinel Di Pesisir Timur Provinsi Lampung”* dalam penelitian ini hasil penelitian yang dilakukan hasil analisis citra senyinel dengan menggunakan transformasi indeks perbedan vegetasi, SAVI (*Soil Adjusted Vegetation Index*) dan NDWI (*Normalized Difference Water Index*) untuk menentukan kerapatan vegetasi. Pada transformasi NDVI yang didominasi kelas kerapatan tinggi, rentang nilainya adalah 0,67 hingga 1 seluas 46975,96 ha, dan pada transformasi SAVI yang didominasi kelas kerapatan sangat jarang, rentang nilainya adalah 0,99 hingga 1,38, seluas 48775,96 ha.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini : (Mubin et al., 2019) dalam jurnal yang berjudul *“Young and Mature Oil Palm Tree Detection and Counting Using Convolutional Neural Network Deep Learning Method”* dalam penelitian ini Penghitungan kelapa sawit mempunyai peranan penting dalam pengelolaan perkebunan. Pengamatan manual tidak akurat, tidak efisien, dan tidak praktis di perkebunan besar, dan lain-lain pendekatan yang digunakan dapat menghasilkan positif palsu dan negatif palsu karena satu kerangka kerja digunakan untuk mendeteksi minyak sawit tua dan muda. Pendekatan penginderaan jauh terbukti berhasil menjadi pendekatan yang lebih baik, namun keterbatasan saat ini mencakup kurangnya penelitian yang membangun sistem deteksi terpisah untuk kelapa sawit muda dan tua, kurangnya

pendekatan yang digunakan pembelajaran mendalam, serta kurangnya penelitian yang menyertakan teknologi GIS dalam pendeteksiannya dan proses penyimpanan data. Penelitian ini berfungsi untuk menutup kesenjangan penelitian tersebut dengan menggunakan metode *deep learning* untuk deteksi dan penghitungan minyak sawit.

2.2 Pengolahan Citra

Gambar pada bidang dua dimensi disebut citra, jika di lihat dari sudut pandang matematis, citra adalah fungsi beulang dari intensitas Cahaya pada bidang dua dimens. Ada dua jenis gambar: gambar diam (sebuah gambar tinggal yang tidak bergerak) dan gambar bergerak (sebuah rangkaian gambar dian yang di tampilkan secara beurutan) memberikan kesan kepada mata sebagai gambar yang bergerak. Jenis gambar termasuk transparansi, gambar berwarna yang dimodelkan kedalam warna RGB (merah, hijau dan biru), gambar *grayscale* (keabu-abuan), yang memiliki nilai normal antara 0 dan 255 (hitam) dan terindeks dengan 16 juta pilihan warna. Pengolahan Citra Digital (*Digital Image Processing*) Pengolahan citra adalah cabang ilmu Informatika untuk memperbaiki kualitas citra agar kualitas lebih baik, lebih mudah di interpetasikan manusia maupun komputer. Deteksi Objek (*Edge Detection*) pendeteksian tepi digunakan untuk mengidentifikasi diskontinuitas *grey level* dan untuk segmentasi atau identifikasi objek. Tujuan pendeteksi tepi adalah untuk menentukan garis tepi suatu objek pada gambar melalui pada gambar melalui pengamatan perubahan tingkat kecerahan yang signifikan atau diskontinuitas. Nilai piksel yang melebihi tepi atau ambang batas, disebut diskontinuitas local. Metode *thresholding*, yang didasarkan pada keabuan

gambar, digunakan untuk segmentasi gambar. Nilai lebih ambang (1) akan diubah menjadi putih (0) dan nilai intensitas gambar yang lebih rendah akan diubah menjadi hitam (0). Gambar menunjukkan bagaimana sumbu sistem berwarna menjadi bagian tertentu (Pratama et al., 2022).

2.2.1 NDVI (Normalized Difference Vegetation Index)

Untuk menentukan tingkat kehijauan, *Normalized Difference Vegetation Index* (NDVI) adalah perhitungan citra. Nilai NDVI dapat digunakan untuk menunjukkan parameter yang berkaitan dengan vegetasi seperti biomasa dan area dedaunan hijau. Nilai indeks berkisar dari nol hingga satu, tetapi masing-masing objek memiliki nilai indeks yang berbeda ketika dibagi, tempat-tempat yang berbunga memiliki rentang antara 0,2 dan 0,8, sedangkan tempat-tempat yang tidak berbunga memiliki antara -1 dan 0 (Pangestu & Banowati, 2023).

Metode yang dapat digunakan dalam pengolahan data penginderaan jauh adalah analisis indeks vegetasi atau NDVI yaitu suatu metode untuk menilai tingkat kerapatan vegetasi menggunakan data gambar landsat 8. Hasil analisis NDVI dapat menghasilkan informasi berupa perbandingan tingkat kehijauan vegetasi menggunakan data gambar satelit sebagai indikator biomasa, dan mencari tahu tingkat kehijauan relative (Lufilah et al., n.d.). Persamaan untuk menghitung NDVI adalah sebagai berikut Gambar 2.1 berikut menampilkan rumus menghitung NDVI.

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Gambar 2.1 Rumus Menghitung NDVI

Keterangan :

NDVI : *Normalized Difference Vegetation Index*

NIR : Nilai spektral saluran *Near Infra Red*

RED : Nilai spektral saluran *Red*

2.2.2 Tree Counting

Proses perhitungan objek digunakan untuk mengetahui jumlah objek dalam cepat berdasarkan hasil otomatis dari ekstrasi fitur. Proses ini dapat digunakan berdasarkan konektivitas objek terhadap piksel sekitarnya, seperti empat piksel atau delapan piksel. Penelitian ini dilakukan pada objek pohon kelapa sawit dengan menggunakan data foto. Salah satu algoritma yang digunakan adalah template mechine. Algoritma ini memungkinkan kita untuk menemukan bagian tertentu dari gambar masukan yang sesuai dengan template yang telah dibuat. Kemudian, algoritma ini menggunakan nilai ambang dalam proses persamaan template untuk menghasilkan jumlah perhitungan pohon kelapa sawit (Mabrur, 2021).

2.3 Bahasa Pemrograman

Dalam jurnalnya, Saragih menyatakan bahwa Bahasa pemrograman adalah prosedur umum untuk mengarahkan *computer* untuk menjalankan perintah tertentu. Bahasa pemrograman adalah kumpulan aturan sintaks dan semantic yang digunakan untuk mendefinisikan program komputer, cara data di simpan dan dikirim dan jenis langkah apa yang dilakukan dalam berbagai situasi (Saragih, 2018).

2.3.1 Vuejs

Vue.js yang juga dikenal sebagai vvue atau viu adalah framework JavaScript progresif yang digunakan untuk membuat tampilan UI berdasarkan arsitektur MVC (*Model, View Controller*). Vue.js adalah proyek open-source yang dilisensikan MIT yang dibuat oleh Evan You pada tahun 2014. *System Reactive Data Binding* adalah fitur Vue.js yang memungkinkan data dan DOM (*Document Object Model*) terikat satu sama lain (Rohayati Sri et al., 2021).

2.3.2 Python

Python adalah Bahasa pemrograman interpretative yang dapat digunakan untuk berbagai tujuan dan filosofi perancangannya terpusat pada tingkat keterbacaan kode. Python digambarkan sebagai bahasa yang digambarkan sebagai Bahasa yang memiliki banyak kemampuan dan kemampuan Bersama dengan sintaksis kode yang sangat jelas. Selain itu memiliki banyak fitur Pustaka standar yang luas dan komprehensif komunitas besar mendukung python. Python umumnya

mendukung berbagai paradigma pemrograman, tetapi juga mendukung pemrograman veorientasi objek, pemrograman impreatif dan pemrograman fungsional. Python biasanya digunakan sebagai bahasa script, seperti halnya bahasa pemrograman dinamins lainnya. Namun, penggunaan bahasa ini lebih luas mencakup aplikasi yang biasanya tidak dilakukan dengan bahasa script (Syahrudin & Kurniawan, 2018).

Berikut adalah beberapa fitur utama dari Python:

1. Sintaks yang Mudah: Python menggunakan sintaks yang sederhana dan mudah dibaca sehingga mengurangi kesalahan penulisan kode dan memungkinkan pengembang untuk lebih fokus pada solusi masalah.
2. Serba guna kesepakatan (*Batteries-Included*): Python dilengkapi dengan berbagai pustaka (*libraries*) standar yang luas dan berguna, yang memungkinkan pengembang untuk menyelesaikan berbagai tugas tanpa harus menulis kode dari nol.
3. Berorientasi Objek: Python mendukung pemrograman berorientasi objek, yang memungkinkan pemodelan data dan fungsionalitas dalam bentuk objek, yang berguna untuk pengorganisasian dan pemeliharaan kode yang lebih baik.
4. Interpretatif dan Dinamis: Python adalah bahasa pemrograman interpretatif, yang berarti kode dapat dieksekusi langsung tanpa perlu dikompilasi terlebih dahulu. Selain itu, Python adalah bahasa pemrograman dinamis juga, yang artinya variabel tidak perlu dideklarasikan dengan tipe data tertentu.

5. Portabilitas: python dapat digunakan pada banyak platform, seperti windows, macOS, linux dan sistem operasi lainnya.
6. Fleksibilitas: Python dapat digunakan untuk berbagai jenis aplikasi, mulai dari pengembangan web, pemrograman server, analitik data, kecerdasan buatan (AI), hingga scripting dan otomatisasi tugas.

2.3.3 Javascript

Framework JavaScript adalah kerangka kerja yang siap digunakan dan sistematis dengan bantuan sejumlah library. Membantu pengembangan membuat situs web berbasis Java Script, seperti membuat fungsi, prosedur, serta fitur lainnya, dengan menggunakan library yang sudah ada sehingga mereka tidak perlu menulis kode sumber lagi. Javascript memiliki dua tipe, yaitu: Framework Javascript Frontend dan Framework Javascript Backend. Framework JavaScript Frontend membantu mengembangkan tampilan antarmuka situs web yang interaktif yang menarik dan meningkatkan pengalaman pengguna. Sebaliknya Framework JavaScript Backend berfokus pada sistem internal seperti pemrosesan data atau processing, parsing data atau parse data, suplai data, penanganan data atau handle permintaan, routing dan peningkatan pengalaman pengguna (Christopher et al., 2022).

2.3 UML

Untuk menyelesaikan banyaknya penggunaan bahasa “pemrograman berorientasi objek” (OOP), UML (*Unified Modeling Language*) adalah metode

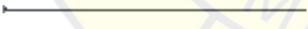
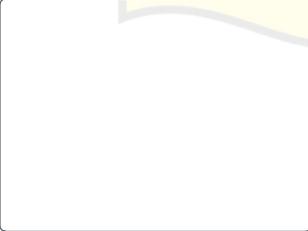
kolaborasi antara metode booth, OMT (*Object Modeling Technigue*), dan OOSE (*Objek Oriebtid Engineering*). Berikut adalah beberapa jenis UML yang digunakan untuk memodelkan sistem perangkat lunak.

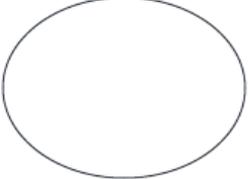
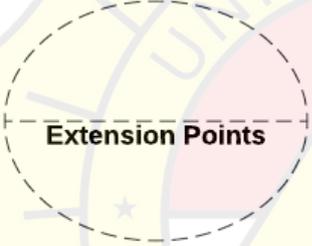
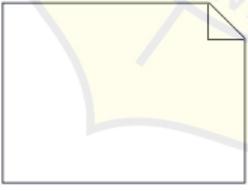
2.4.1 Usecase Diagram

Diagram yang menggambarkan aktor dan fungsinya dalam hubungannya sebagai urutan tindakan yang memberi nilai terukur kepada aktor dalam diagram UML usecase, sebuah usecase diwakili dengan elips horizontal. Tabel 2.1 menunjukkan notasi usecase diagram.

Tabel 2.1 Usecase Diagram

| Gambar | Nama | Keterangan |
|---|-------------------|--|
|  | Aktor | Menentukan peran pengguna saat menangani usecase. |
|  | <i>Dependency</i> | Suatu hubungan dimana transformasi terjadi pada anggota independen mempengaruhi anggota yang bergantung pada |

| Gambar | Nama | Keterangan |
|---|-----------------------|---|
| | | anggota non-independen |
|  | <i>Generalization</i> | Ketika objek anak (descendent) berbagi perilaku dan struktur data dalam objek induk (ancestor). |
|  | <i>Include</i> | Menentukan contoh sumber. |
|  | <i>Extend</i> | Menentukan bahwa usecase target dapat memperluas perilaku usecase sumber pada titik tertentu. |
|  | <i>Association</i> | Menghubungkan dua objek. |
|  | <i>System</i> | Menyediakan paket dengan sistem terbatas. |

| Gambar | Nama | Keterangan |
|---|----------------------|---|
|  | Usecase | Gambaran tentang urutan tindakan yang ditampilkan oleh sistem yang menghasilkan hasil dapat diukur bagi suatu aktor. |
|  | <i>Collaboration</i> | Kombinasi aturan dalam elemen lain yang berkerja sama untuk menghasilka prilaku yang lebih besar dari jumlah dan elemennya (sinergi). |
|  | <i>Note</i> | Elemen fisik yang muncul saat aplikasi berjalan dan menunjukan sumber data komputasi. |

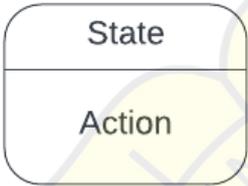
2.4.2 Activity Diagram

Diagram Activity dikenal juga diagram aktivitas adalah cara utama untuk menjelaskan logika procedural, proses bisnis, dan aliran bisnis dan aliran kerja

dalam kasus. Activity diagram memiliki perah seperti halnya diagram alir mendukung perilaku parallel, sedangkan diagram aliran tidak bisa. Tabel 2.2 menunjukkan notasi diagram activity.

Tabel 2.2 Activity Diagram

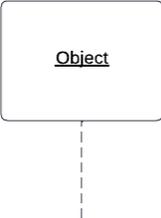
| Gambar | Nama | Keterangan |
|---|----------------------|---|
|  | Status Awal | Status awal ditampilkan pada diagram activity. |
|  | Status Akhir | Sebuah digram activity menunjukkan status akhir dari langkah terakhir yang diambil sistem. |
|  | Pembangunan/Join | Pembangunan dimana yang mana telah terjadi lebih dari satu tindakan sebelumnya digabungkan menjadi satu |
|  | Percabangan/Decision | Percabangan dimana aktivitas melibatkan lebih dari satu. |

| Gambar | Nama | Keterangan |
|---|-----------------|--|
|  | Aktivitas | Semua aktivitas yang dilakukan oleh sistem biasanya dimulai dengan kata kerja. |
|  | <i>Swimlane</i> | Swimlane membedakan organisasi bisnis yang bertanggung jawab atas tindakan yang terjadi. |

2.4.3 Sequence Diagram

Merupakan diagram yang menunjukkan fungsionalitas usecase. Sequence diagram adalah salahsatu dari dua indikasi diagram yang menunjukkan objek yang terkait dengan usecase dan message atau pesannya. Komponen utama sequence diagram terdiri dari atas pbjek yang dituliskan dengan kotak segitiga berwarna. Berita diwakili oleh garis tanda panah dan waktu progress vertical. Tabel 2.3 menunjukkan notasi Sequence Diagram.

Tabel 2.3 Sequence Diagram

| Gambar | Nama | Keterangan |
|---|----------------------|---|
|  | Objek | Komponen yang membuat sebuah objek dalam diagram. |
|  | <i>Stimulus</i> | Untuk menunjukkan bagaimana komunikasi antara objek berinteraksi. |
|  | <i>Focus Control</i> | Sebagai lokasi untuk input dan output dari proses yang dilakukan oleh objek atau aktor yang disistem. |
|  | <i>Self Stimulus</i> | Meskipun fungsional mirip dengan stimulus, pesan yang disampaikan |

| Gambar | Nama | Keterangan |
|--------|------|---------------------------------------|
| | | ditunjukkan kepada objek itu sendiri. |

2.5 Qgis

QGIS adalah perangkat Sistem Informasi Geografis (GIS) open-source yang mudah digunakan yang dilisensikan oleh GNU *General Public License*. QGIS berjalan pada linux, Unixn, Mac OSX, Windows dan Android dan mendukung berbagai format vector, raster, dan database fitur (Pribadi & Hastarini, 2022).

2.6 VSCode

Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang tersedia untuk sistem operasi desktop, Windows, macOS, dan Linux. Ada dukungan built-in untuk JavaScript, TypeScript dan Node.js, serta ekosistem ekstensi yang luas untuk bahasa laub (seperti C++, C#, Java, Python, Php, Go) dan runtime (seperti.NET dan Unity) (Romzi & Kurniawan, 2020).