BAB II
TINJAUAN PUSTAKA

2.1 Potensi Sumber Daya Perikanan di Indonesia

Berdasarkan Keputusan Menteri Kelautan dan Perikanan Nomor 19 Tahun 2022 mengenai Estimasi Potensi, Jumlah Tangkapan yang Diperbolehkan, serta Tingkat Pemanfaatan Sumber Daya Ikan di WPPNRI, potensi lestari sumber daya ikan nasional diperkirakan mencapai 12.011.125 ton per tahun, yang mencakup berbagai jenis komoditas perikanan laut. Adapun jenis perikanan laut meliputi: ikan pelagis kecil, ikan pelagis besar, ikan demersal, ikan karang, udang penaeid, lobster, kepiting, rajungan, dan cumi-cumi. Berdasarkan data dari Kementerian Kelautan dan Perikanan (2024), total produksi perikanan tangkapan laut sebesar 6.809.913 ton. Total tangkapan laut tersebut diperoleh dari aktivitas penangkapan di WPPNRI.

2.2 Wilayah Pengelolaan Perikanan Negara Republik Indonesia (WPPNRI)

Berdasarkan Peraturan Menteri Kelautan dan Perikanan Republik Indonesia Nomor.18/PERMEN-KP/2014 dalam Pasal 2 mengenai WPPNRI dibagi dalam 11 wilayah pengelolaan perikanan yaitu:

1. WPPNRI 571 meliputi perairan Selat Malaka dan Laut Andaman,
2. WPPNRI 572 meliputi perairan Samudera Hindia sebelah Barat Sumatera dan Selat Sunda,
3. WPPNRI 573 meliputi perairan Samudera Hindia sebelah Selatan Jawa hingga sebelah Selatan Nusa Tenggara, Laut Sawu, dan Laut Timor bagian Barat,
4. WPPNRI 711 meliputi perairan Selat Karimata, Laut Natuna, Laut China Selatan,
5. WPPNRI 712 meliputi perairan Laut Jawa,
6. WPPNRI 713 meliputi perairan Selat Makassar, Teluk Bone, Laut Flores, dan Laut Bali,
7. WPPNRI 714 meliputi perairan Teluk Tolo dan Laut Banda,
8. WPPNRI 715 meliputi perairan Teluk Tomini, Laut Maluku, Laut Halmahera, Laut Seram, dan Teluk Berau,

9. WPPNRI 716 meliputi perairan Laut Sulawesi dan sebelah Utara Pulau Halmahera,
10. WPPNRI 717 meliputi perairan Cendrawasih dan Samudera Pasifik,
11. WPPNRI 718 meliputi perairan Laut Aru, Laut Arafuru, dan Laut Timor bagian Timur.

2.3 Ikan Pelagis

Ikan pelagis merupakan kelompok ikan yang hidup di lapisan permukaan hingga kolom air, dengan ciri khas utama yaitu selalu bergerak dalam kelompok *(schooling)* dan melakukan migrasi untuk memenuhi kebutuhan hidupnya. Ikan pelagis dibagi menjadi dua berdasarkan ukuran tubuhnya, yaitu ikan pelagis besar seperti tuna, cakalang, dan tongkol, serta ikan pelagis kecil seperti layang, teri, dan kembung (Nelwan, 2004). Habitat ikan pelagis di pengaruhi oleh berbagai faktor lingkungan perairan, terutama aspek fisik dan kimia. Parameter fisik dan kimia perairan sering digunakan untuk mengamati tingkat kelimpahan ikan. Kondisi oseanografi seperti salinitas, kecepatan arus, SPL, konsentrasi klorofil-a, dan karakteristik kedalaman laut mencerminkan parameter tersebut (Pratama et al., 2024). Dari berbagai parameter oseanografi tersebut, SPL menjadi faktor yang paling menonjol karena berpengaruh langsung terhadap metabolisme ikan dan distribusi habitatnya. Oleh sebab itu, SPL sering dijadikan indikator utama dalam kajian potensi penangkapan ikan (Siregar et al., 2015).

2.4 Penginderaan Jauh

Penginderaan jauh adalah proses memperoleh data atau informasi tentang objek atau fenomena tanpa kontak langsung menggunakan sensor yang menghasilkan citra sebagai hasil rekaman. Salah satu tahap pentingnya adalah interpretasi citra yaitu mengenali bentuk dan sifat objek yang tampak, yang dapat dilakukan secara manual maupun digital. Metode manual melibatkan identifikasi objek pada citra yang telah terkoreksi menggunakan unsur interpretasi citra, skala, dan resolusi, sedangkan metode digital memanfaatkan komputer untuk pra-pengolahan, penajaman, hingga klasifikasi. Jika citra digital sudah terkoreksi, pengguna cukup melakukan klasifikasi dan koreksi geometris untuk meningkatkan ketelitian hasil interpretasi (Putra et al., 2023).

Tujuan penginderaan jauh adalah memperoleh data dan informasi dari citra, baik foto maupun nonfoto, mengenai berbagai objek di permukaan bumi yang direkam atau digambarkan menggunakan sensor buatan. Pemahaman tentang dasar-dasar interpretasi penginderaan jauh menjadi keterampilan awal yang penting sebelum melakukan interpretasi citra, baik foto maupun nonfoto dalam berbagai bidang (Yusuf, 2017).

2.5 Suhu Permukaan Laut (SPL)

SPL merupakan salah satu variabel utama dalam memahami serta memprediksi pertukaran panas, momentum, dan gas antara laut dan atmosfer. SPL berperan penting terhadap dinamika cuaca dan iklim, baik dalam skala global melalui sistem arus laut dan sirkulasi atmosfer, maupun dalam skala lokal seperti pembentukan angin laut dan awan (O'Carroll et al., 2019). Pemantaun SPL penting dalam studi oseanografi dan perikanan karena dapat mencerminkan kondisi lingkungan laut yang memengaruhi distribusi dan kelimpahan sumber daya ikan.

Produktivitas perairan Asia Tenggara dipengaruhi oleh ketersediaan nutrien yang mengalami siklus pengayaan kembali secara musiman melalui proses *upwelling*. *Upwelling* adalah proses oseanografi yang dipicu oleh divergensi horizontal akibat angin di permukaan laut. Divergensi ini menyebabkan air dari lapisan bawah bergerak naik untuk menggantikan massa air permukaan yang hilang, sehingga menghasilkan aliran vertikal ke atas dalam kolom perairan. Keberadaan massa air dingin dari lapisan bawah yang berdampingan dengan perairan permukaan yang lebih hangat menghasilkan gradien suhu horizontal, yang secara fisik membentuk suatu *thermal front.* Kondisi ini menunjukkan bahwa dinamika oseanografi di perairan Indonesia erat kaitannya dengan variasi musiman, yang selanjutnya memengaruhi sebaran SPL, konsentrasi klorofil-a, dan potensi penangkapan ikan pelagis (Wyrtki, 1961; Kampf & Chapman, 2016; Stewart, 2008). Variasi spasial dan temporal SPL tidak hanya menggambarkan dinamika pemanasan laut, tetapi juga mengindikasikan adanya batas *thermal front.*

Thermal front merupakan wilayah pertemuan dua massa air yang memiliki perbedaan SPL yang tajam. Wilayah ini umumnya memiliki produktivitas yang tinggi karena berfungsi sebagai tempat berkumpulnya unsur hara dari kedua massa

air yang berinteraksi, sehingga menjadi daerah mencari makan (*feeding ground)* bagi ikan pelagis. Kondisi tersebut menjadikan wilayah ini sebagai lokasi potensi untuk penangkapan ikan pelagis (Arief, 2004). Pengolahan data spasial dalam mendeteksi *thermal front* menggunakan metode SIED yang membagi *thermal front* menjadi dua kategori, yaitu *thermal front* lemah dengan perbedaan SPL sebesar $0{,}3°\text{C} - 0{,}49°\text{C}$ dan *thermal front* kuat dengan perbedaan SPL $\geq 0{,}5°\text{C}$ (L Ahmad et al., 2017). Beberapa penelitian mengidentifikasi bahwa *thermal front* umumnya terbentuk pada wilayah dengan perbedaan SPL sekitar 0,5°C (Hamzah et al., 2014;Jatisworo et al., 2013).

2.6 Metode *Single Image Edge Detection* (SIED)

Metode *Single Image Edge Detectio*n (SIED) dikembangkan oleh Cayula & Cornillon (1992) untuk mendeteksi *thermal front* secara otomatis pada citra SPL tunggal. Metode ini berbasis analisis histogram dan segmentasi statistika, dengan prinsip utama mendeteksi distribusi bimodal yang menunjukkan keberadaan dua massa air yang berbeda berdasarkan suhu. Algoritma ini bekerja melalui tiga tahap utama:

a. Analisis Histogram

Dalam tahap awal, citra dibagi menjadi jendela berukuran $16 \times 16, 32 \times 32$, atau 64×64 piksel. Algoritma mengasumsikan bahwa data pada jendela analisis terdiri atas dua kelompok populasi. Kelompok ω_1 didefinisikan sebagai populasi dingin dan kelompok ω_2 membentuk populasi hangat. Misalkan x adalah sebuah sampel (piksel) dari himpunan χ (jendela yang sedang dianalisis), $t(x)$ adalah suhu dari sampel x dan $h(t)$ adalah frekuensi suhu dari sampel x. Setelah dua kelompok populasi ditentukan dengan ambang batas (τ), parameter rata-rata masing populasi (μ) dapat dihitung dengan mudah:

$$\mu_1(\tau) = \frac{\sum_{t<\tau} t \times h(t)}{\sum_{t<\tau} h(t)}, \quad (1)$$

$$\mu_2(\tau) = \frac{\sum_{t\geq\tau} t \times h(t)}{\sum_{t\geq\tau} h(t)}. \quad (2)$$

Selanjutnya, kriteria $\theta(\tau_{opt})$ digunakan untuk menentukan jumlah populasi data, yang didefinisikan sebagai pembagian antara *between-cluster variance* $J_b(\tau)$ dan *total variance* (S_{tot}). Secara teoritis, kriteria tersebut

ditentukan berdasarkan nilai $\theta(\tau_{opt})$, yaitu data dikatakan membentuk dua populasi apabila $\theta(\tau_{opt}) \geq 0{,}7$, sedangkan apabila $\theta(\tau_{opt}) < 0{,}7$, maka data dikategorikan sebagai satu populasi. Nilai ambang 0,7 tersebut ditetapkan secara empiris berdasarkan keseluruhan hasil simulasi yang dilakukan oleh Cayula & Cornillon (1992) dan digunakan sebagai batas minimum separabilitas dalam membedakan distribusi bimodal dan unimodal. *Total variance* (S_{tot}) merupakan penjumlahan dari *within-cluster variance* $J_e(\tau)$ dan *between-cluster variance* $J_b(\tau)$, sehingga dapat dinyatakan sebagai:

$$S_{tot} = J_e(\tau) + J_b(\tau). \quad (3)$$

Within-cluster variance dirumuskan sebagai berikut:

$$J_e(\tau) = \frac{N_1}{N_1+N_2} S_1(\tau) + \frac{N_2}{N_1+N_2} S_2(\tau), \quad (4)$$

dengan:

N_1 = jumlah data pada populasi dingin,
N_2 = jumlah data pada populasi hangat,
$S_1(\tau)$ = varians data pada populasi dingin,
$S_2(\tau)$ = varians data pada populasi hangat.

Pada Persamaan (4) , $S_1(\tau)$ dan $S_2(\tau)$ didefinsikan sebagai:

$$S_1(\tau) = \left(\frac{\sum_{t<\tau}[t-\mu_1(\tau)]^2 h(t)}{N_1}\right), \quad (5)$$

$$S_2(\tau) = \left(\frac{\sum_{t\geq\tau}[t-\mu_2(\tau)]^2 h(t)}{N_2}\right). \quad (6)$$

Sedangkan *between-cluster variance* dirumuskan sebagai:

$$J_b(\tau) = \frac{N_1 N_2}{(N_1+N_2)^2}[\mu_1(\tau) - \mu_2(\tau)]^2, \quad (7)$$

dengan:

N_1 = jumlah data pada populasi dingin,
N_2 = jumlah data pada populasi hangat,
$\mu_1(\tau)$ = nilai rata-rata data pada populasi dingin,
$\mu_2(\tau)$ = nilai rata-rata data pada populasi hangat.

b. Algoritma Kohesi

Algoritma pada tahap awal hanya melihat distribusi nilai data. Namun, distribusi yang terlihat memiliki dua kelompok belum tentu mencerminkan

keberadaan dua kelompok yang terpisah secara spasial. Hal ini bisa disebabkan oleh gangguan pada data. Oleh karena itu, kedua kelompok diuji lebih lanjut untuk memastikan koherensi spasialnya. Dalam perhitungan ini, hanya empat tetangga terdekat yaitu $\mathcal{N}(x_{i,j}) = \{x_{i,j+1}, x_{i,j-1}, x_{i+1,j}, x_{i-1,j}\}$ yang dipertimbangkan. Populasi $\omega_1' \; dan \; \omega_2'$ didefinisikan sedemikian rupa sehingga, untuk sebuah piksel x dengan suhu $t(x)$:

$$t(x) \leq \tau_{opt} \rightarrow \omega_1' \text{ dan } t(x) > \tau_{opt} \rightarrow x \in \omega_2'. \quad (8)$$

Koefisien kohesi untuk populasi $\omega_1' \; dan \; \omega_2'$, didefinisikan sebagai berikut:

$$C_1 = \frac{R_1}{T_1}, \quad (9)$$

$$C_2 = \frac{R_2}{T_2}, \quad (10)$$

$$C = \frac{R_1 + R_2}{T_1 + T_2}, \quad (11)$$

dimana T_1, yaitu jumlah total pasangan antara piksel pusat yang termasuk dalam populasi dingin $(\omega_1{}')$ dan piksel tetangga yang termasuk dalam salah satu populasi, diberikan oleh:

$$T_1 = |\{(x,y), \text{ sedemikian sehingga } y \in [\mathcal{N}(x) \cap \chi] \; \forall x \in \omega_1'\}|, \quad (12)$$

dan R_1 yaitu jumlah total pasangan antara piksel pusat dan tetangga yang keduanya termasuk dalam populasi dingin $(\omega_1{}')$, diberikan oleh:

$$R_1 = |\{(x,y), \text{sedemikian sehingga } y \in [[\mathcal{N}(x) \cap \omega_1'] \; \forall x \in \omega_1'\}|. \quad (13)$$

Selanjutnya, $R_2 \; dan \; T_2$ didefinisikan dengan cara yang sama, yaitu dengan menggantikan ω_1' dengan ω_2' dalam Persamaan (12) dan (13). Notasi $|\;|$ menyatakan kardinalitas suatu himpunan, yaitu banyaknya elemen dalam himpunan tersebut. Ambang batas untuk menghilangkan tepi yang dihasilkan dari distribusi *noise*, yaitu jika $C < 0{,}92$, $C_1 < 0{,}90$, dan $C_2 < 0{,}90$ (Cayula & Cornillon, 1992).

c. Menentukan Lokasi Piksel Tepi

Algoritma level jendela yang dibahas sebelumnya dirancang untuk mendeteksi dan memastikan keberadaan tepi di setiap jendela individu. Langkah terakhir dalam proses ini adalah menentukan lokasi piksel tepi di jendela-jendela di mana keberadaan *thermal front* telah terdeteksi dan tervalidasi. Output dari

algoritma ini adalah citra tepi. Dengan menggunakan fungsi indikator $\Omega(x)$ sehingga:

$$\forall x \in \chi\ , \Omega(x) = \begin{cases} 0, jika\ x \in \omega_1' \\ 1, jika\ x \in\ \omega_2' \end{cases}. \quad (14)$$

2.7 *ModelBuilder*

ModelBuilder merupakan sebuah bahasa pemrograman visual yang digunakan untuk membangun alur kerja (*workflow*) *geoprocessing* dalam sistem informasi geografis. Model ini direpresentasikan dalam bentuk diagram yang menghubungkan serangkaian data, proses, dan *tools*, di mana output dari suatu proses menjadi input bagi proses berikutnya. Dengan demikian, *ModelBuilder* memungkinkan pengguna untuk mengotomatisasi sekaligus mendokumentasikan proses analisis spasial dan manajemen data secara sistematis (Esri, 2023).

ModelBuilder berfungsi sebagai media interaktif untuk menyusun *workflow* analisis dengan menggabungkan berbagai *tools* analisis yang tersedia. Alur kerja yang dibangun dapat berupa proses sederhana seperti manajemen data hingga analisis spasial yang kompleks. Selain itu, *ModelBuilder* membantu menyederhanakan analisis dengan memungkinkan eksekusi berulang , otomatisasi proses bertahap, serta dokumentasi *workflow* secara visual (Esri, 2022).

Dalam penggunaannya, *ModelBuilder* memungkinkan pengguna untuk membangun model dengan menambahkan dan menghubungkan data serta *tools*, memvalidasi input dan proses, serta menjalankan model baik secara keseluruhan maupun per langkah tertentu. Model yang telah dibuat juga dapat disimpan, dimodifikasi, dan digunakan kembali dengan parameter atau input yang berbeda, bahkan dapat dipublikasikan sebagai alat *geoprocessing* atau diintegrasikan dengan skrip Python (Esri, 2023).

Dalam konteks aplikasi GIS, *ModelBuilder* sering digunakan untuk membangun model analisis berbasis data vektor maupun raster. *ModelBuilder* biasanya dimanfaatkan untuk tugas-tugas yang melibatkan rangkaian proses *geoprocessing*. Sementara itu, untuk kebutuhan yang lebih kompleks atau fleksibel, proses analisis juga dapat dikombinasikan dengan skrip Python, misalnya dalam

proses perhitungan matematis dan logika pada data raster untuk menghasilkan informasi baru (Kang-tsung Chang, 2018).

2.8 *Point Density*

Point density merupakan metode analisis spasial yang digunakan untuk menghitung kepadatan titik dalam suatu wilayah. Metode ini bekerja dengan menghitung jumlah titik yang berada di sekitar setiap sel raster hasil analisis dalam suatu *neighborhood* tertentu. Secara konseptual, suatu *neighborhood* didefinisikan di sekitar pusat setiap sel raster. Jumlah titik yang berada di dalam *neighborhood* tersebut dihitung dan dibagi dengan luas lingkungan sehingga diperoleh nilai kepadatan per satuan luas (ESRI, 2023). Berdasarkan konsep tersebut, kepadatan titik secara matematis dapat dinyatakan sebagai berikut:

$$D = \frac{N}{A}, \tag{15}$$

dengan:

D = nilai kepadatan titik,

N = jumlah titik yang berada dalam *neighborhood,*

A = luas area *neighborhood.*

Hasil dari proses ini berupa data raster yang menunjukkan besaran kepadatan titik pada setiap sel raster dan banyak digunakan untuk menganalisis distribusi spasial objek seperti lokasi bangunan atau kejadian tertentu dalam suatu wilayah (Mohammed et al., 2021).

2.9 Algoritma K-Means

Misalkan suatu himpunan semesta data X terdiri atas n objek yang berada dalam ruang Euclidean. Metode partisi mendistribusikan objek-objek dalam X ke dalam k klaster, yaitu $C_1, \dots, C_k$, dengan ketentuan bahwa $C_i \subset X$ dan $C_i \cap C_j = \emptyset$ untuk setiap $i, j \in \{1,2, \dots, k\}$. Sebuah fungsi objektif digunakan untuk menilai kualitas hasil partisi, sehingga objek-objek di dalam satu klaster memiliki tingkat kemiripan yang tinggi, sedangkan objek-objek pada klaster yang berbeda memiliki tingkat ketidakmiripan yang tinggi. Teknik partisi berbasis *centroid* menggunakan *centroid* dari suatu klaster C_i sebagai representasi klaster tersebut. Secara konseptual, *centroid* merupakan titik pusat dari suatu klaster. Kualitas klaster C_i

ditentukan oleh total kuadrat jarak antara setiap objek dalam klaster tersebut dan *centroid* c_i, yang didefinisikan sebagai (Han et al., 2012):

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist\,(p, c_i)^2, \tag{16}$$

dimana:

C_i = himpunan data pada klaster ke-i,

c_i = *centroid* dari klaster C_i,

p = vektor data yang menjadi anggota klaster C_i,

k = jumlah total klaster,

$dist(p, c_i)$ = jarak Euclidean antara dua titik $p \in C_i$ dan c_i.

Langkah-langkah algoritma K-Means adalah sebagai berikut:

1. Tentukan jumlah klaster,
2. Pilih secara acak *centroid* awal c_i sebanyak jumlah k klaster,
3. Hitung jarak setiap data p terhadap seluruh *centroid* c_i, kemudian tempatkan data tersebut ke dalam klaster C_i yang memiliki jarak paling dekat dengan *centroid*. Jarak antara data p dan *centroid* c_i dihitung menggunakan jarak Euclidean sebagai berikut:

$$dist(p, c_i) = \sqrt{\sum_{z=1}^{n} (p_z - c_{iz})^2}, \tag{17}$$

4. *Centroid* c_i untuk setiap $i \in \{1,2, \dots, k\}$, diperbarui dengan menghitung rata-rata seluruh anggota klaster C_i, yaitu:

$$c_i = \frac{1}{|C_i|} \sum_{p \in C_i} p, \tag{18}$$

5. Ulangi langkah 3 dan 4 hingga *centroid* tidak mengalami perubahan atau tidak terdapat perubahan keanggotaan data dalam klaster.