



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

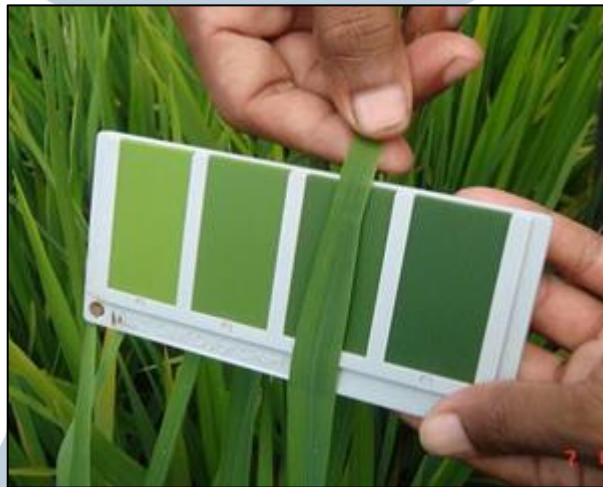
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Bagan Warna Daun (BWD)

Bagan warna daun (BWD) adalah alat berbentuk persegi empat yang berguna untuk mengetahui status hara N tanaman padi. Pada alat ini terdapat empat kotak skala warna, mulai dari hijau muda hingga hijau tua, yang mencerminkan tingkat kehijauan daun tanaman padi. Sebagai contoh, kalau daun tanaman berwarna hijau muda berarti tanaman kekurangan hara N sehingga perlu dipupuk. Sebaliknya, jika daun tanaman berwarna hijau tua atau tingkat kehijauan daun sama dengan warna di kotak skala 4 pada BWD berarti tanaman sudah memiliki hara N yang cukup sehingga tidak perlu lagi dipupuk (Sriyanto, 2010).



Gambar 2.1 Bagan warna daun (Sumber : www.pertanian.go.id)

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Pemupukan N yang kurang dari kebutuhan tanaman akan memberikan hasil panen padi rendah (tidak optimal), sebaliknya pemupukan N berlebih tanaman menjadi lebih peka terserang penyakit dan mudah rebah. Sehingga dengan BWD dapat menekan biaya pupuk 15-20% dari takaran yang umum digunakan petani tanpa menurunkan hasil. Melalui penelitian yang dilakukan, BWD dapat digunakan 2 cara yaitu berdasarkan waktu yang telah ditetapkan (fixed time) atau berdasarkan kebutuhan rill tanaman (www.pertanian.go.id).

Terdapat 2 waktu penggunaan BWD, antara lain (Erythrina, 2016):

a. Berdasarkan waktu yang telah ditetapkan

Biasanya berdasarkan pertumbuhan tanaman, yaitu saat pembentukan anakan aktif pada 21-28 hari setelah tanam (HST) dan primordia pada 35-40 HST. Dengan cara ini hanya perlu dilakukan 2 kali pengukuran warna daun padi dengan BWD, karena pada pemupukan pertama tidak perlu digunakan BWD.

b. Berdasarkan kebutuhan riil tanaman

Dengan membandingkan warna daun padi dengan skala BWD secara berkala, setiap 7-10 hari sejak 21-28 hari setelah tanam (HST) sampai 50 HST. Tanaman segera diberi pupuk N bila warna daun berada di bawah skala 4 BWD. Dengan cara ini petani perlu sering ke sawah membandingkan warna daun padi dengan BWD.

2.1.1 Takaran Pupuk N Berdasarkan Waktu yang Telah Ditetapkan

BWD hanya digunakan pada pemupukan kedua atau stadia anakan aktif (21-28 HST) dan pemupukan ketiga atau primordia (35-40 HST) dengan membandingkan warna daun dengan skala BWD.

- a. Bila warna daun berada pada skala 2 sampai 3, berikan 125 kg urea/ha kalau hasil yang biasa dicapai di suatu tempat 7 t/ha gabah kering giling (GKG). Berikan 75 kg urea/ha kalau tingkat hasil 5 t/ha GKG (Tabel 2.1).
- b. Bila warna daun berada antara skala 3 dan 4, berikan 100 kg urea/ha kalau hasil yang biasa dicapai adalah 7 t/ha GKG. Cukup berikan 50 kg urea/ha kalau tingkat hasil adalah 5 t/ha GKG.
- c. Bila warna daun pada skala 4 sampai 5, berikan 50 kg urea/ha kalau hasil yang biasa dicapai 7-8 t/ha GKG. Tanaman tidak perlu dipupuk N kalau tingkat hasil adalah 5-6 t/ha.

Tabel 2.1 Takaran urea yang diberikan sesuai dengan skala warna daun pada penggunaan BWD berdasarkan waktu yang telah ditetapkan. (Sumber : www.pertanian.go.id)

Nilai warna daun dengan BWD	Tingkat hasil (t/ha GKG)			
	5	6	7	8
	Takaran Urea yang digunakan (kg/ha)			
2 – 3	75	100	125	150
Antara 3 dan 4	50	75	100	125
4 – 5	0	0-50	50	50

Tingkat hasil pada kondisi kebutuhan tanaman akan unsur hara lain seperti P dan K terpenuhi serta faktor lain seperti pengendalian organisme pengganggu tanaman (OPT) dan pengelolaan air dilakukan secara optimal.

2.1.2 Takaran Pupuk N Berdasarkan Kebutuhan Riil Tanaman

- a. Pengukuran warna daun padi dengan BWD dimulai pada 21-28 HST, dilanjutkan setiap 7-10 hari sekali sampai 50 HST.
- b. Apabila tingkat hasil di suatu tempat sebesar 7 t/ha GKG, takaran pupuk urea susulan yang diperlukan adalah 100 kg/ha. Bila tingkat hasil adalah 5 t/ha GKG, cukup diberikan 50 kg urea/ha (Tabel 2.2).

Tabel 2.2 Takaran urea susulan yang diperlukan bila warna daun di bawah nilai kritis (skala <4 BWD) pada penggunaan BWD berdasarkan kebutuhan riil tanaman. (Sumber : www.pertanian.go.id)

Nilai warna daun dengan BWD	Tingkat hasil (t/ha GKG)			
	5	6	7	8
	Takaran Urea yang digunakan (kg/ha)			
Di bawah 4	50	75	100	125

Tingkat hasil pada kondisi kebutuhan tanaman akan unsur lain seperti P dan K terpenuhi serta faktor lain seperti pengendalian organisme pengganggu tanaman (OPT) dan pengelolaan air dilakukan secara optimal.

2.2 Citra Digital

Citra merupakan keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada kawasan waktu atau bidang dan untuk menyatakan nilai keabuan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra, yaitu cara kontinu-kontinu, kontinu-diskret, diskret-kontinu, dan diskret-diskret (Santi, 2011).

Pada komputer digital bekerja dengan angka presisi berhingga dengan demikian hanya citra dari kelas diskret-diskret yang dapat diolah dengan komputer. Citra dari kelas tersebut lebih dikenal sebagai citra digital, citra digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Jadi informasi yang terkandung bersifat

diskret, namun citra digital tidak selalu merupakan hasil langsung data rekaman suatu sistem. Kadang-kadang hasil rekaman data bersifat kontinu seperti gambar pada monitor televisi, foto, sinar, dan lain sebagainya. Dengan demikian untuk mendapatkan suatu citra digital diperlukan suatu proses konversi, sehingga citra tersebut selanjutnya dapat diproses dengan komputer (Santi, 2011).

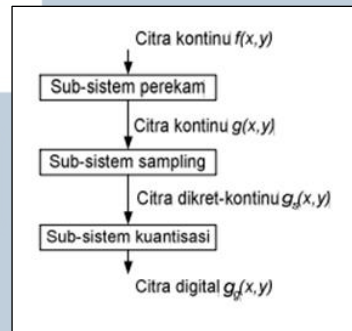
Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mangacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah *array* yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Putra, 2010).

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai amplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital (Putra, 2010).

2.2.1 Sistem Penangkap Citra Digital

Citra Digital dapat diperoleh secara otomatis dari sistem penangkap citra digital (*digital image acquisition system* atau *digitizer*) yang melakukan penjelajahan citra dan membentuk suatu matriks dimana elemen-elemennya menyatakan nilai intensitas cahaya pada suatu himpunan diskret dari titik-titik. Sistem tersebut merupakan bagian terdepan dari suatu sistem pengolah citra. Sistem penangkap citra digital sendiri terdiri dari tiga komponen dasar, yaitu: pengindra

citra yang bekerja sebagai pengukur intensitas cahaya, perangkat penjelajah yang bertugas merekam hasil pengukuran intensitas pada seluruh bagian citra, dan pengubah analog ke digital yang mengubah harga kontinu menjadi harga diskret sehingga dapat diproses dengan komputer. Diagram sistem penangkapan citra dapat dilihat pada Gambar 2.1 (Santi, 2011).



Gambar 2.2 Diagram sistem penangkap citra

Citra $f(x,y)$ disimpan dalam memori komputer atau penyimpan bingkai citra dalam bentuk larik $N \times M$ dari sampel diskret dengan jarak sama seperti yang ditunjukkan oleh Gambar 2. Setiap elemen dari larik pada Gambar 2 disebut sebagai elemen citra atau piksel, yang merupakan suatu daerah persegi empat kecil dengan ukuran tertentu, ukuran piksel ini sering disebut sebagai resolusi piksel (Santi, 2011).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N,0) & f(N,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Gambar 2.3 Larik dari piksel

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.3 Konsep Algoritma Histogram of s-RGB

Menurut Mudjirahardjo, dkk (2016), bahwa langkah pertama algoritma *s-RGB* dalam Mudjirahardjo, dkk (2016) adalah menghitung jumlah intensitas RGB (*s-RGB*) pada setiap piksel. Perhitungan dapat dilakukan menggunakan rumus (1).

$$s-RGB(x, y) = I_R(x, y) + I_G(x, y) + I_B(x, y) \dots\dots\dots(1)$$

Dimana *s-RGB* (*x, y*) adalah jumlah dari intensitas *RGB* pada piksel (*x, y*), $I_R(x, y)$, $I_G(x, y)$, dan $I_B(x, y)$ adalah intensitas merah, hijau dan biru di masing-masing piksel (*x, y*). Ketika menggunakan 8 bit untuk kode intensitas warna, maka mendapatkan nilai *s-RGB* dari 0-765. Lalu nilai *s-RGB* tersebut dibagi menjadi 16 *bin* dan dibuatlah histogram *s-RGB* yang menunjukkan frekuensi nilai dari *s-RGB*. Selanjutnya menentukan modus determinan biner dengan rumus (2).

$$\text{mod}_{s-RGB} = \arg \max(\text{histogram}_{s-RGB}) \dots\dots\dots(2)$$

Setelah mendapatkan nilai modus, untuk setiap piksel yang memiliki nilai *s-RGB* dalam *bin* modus akan diproses untuk membuat histogram dari intensitas untuk setiap intensitas *RGB*. Histogram ini menunjukkan frekuensi kemunculan intensitas warna. Untuk menentukan warna dominan dalam modus *bin* yaitu menghitung modus warna dari masing-masing histogram dari intensitas seperti rumus berikut ini.

$$\text{mod}_{\text{hist}_R} = \arg \max_{I_R}(\text{histogram}_R) \dots\dots\dots(3)$$

$$\text{mod}_{\text{hist}_G} = \arg \max_{I_G}(\text{histogram}_G) \dots\dots\dots(4)$$

$$\text{mod}_{\text{hist}_B} = \arg \max_{I_B}(\text{histogram}_B) \dots\dots\dots(5)$$

Selanjutnya dominan warna (mod_{RGB}), dapat dihitung menggunakan rumus (6).

$$\text{mod}_{\text{RGB}} = \max(\text{mod}_{\text{hist}_R}, \text{mod}_{\text{hist}_G}, \text{mod}_{\text{hist}_B}) \dots\dots\dots(6)$$

Setelah mendapat warna dominan, tentukan identitas dari piksel seperti pada rumus (7).

$$I_p = \begin{cases} 1 & \text{if } s\text{-RGB within } \text{mod}_{s\text{-RGB}} \\ 0 & \text{otherwise} \end{cases} \dots\dots\dots(7)$$

Lalu, tentukan apakah piksel di dalam citra merupakan *background* atau *foreground* dengan menggunakan rumus (8).

$$\text{piksel} = \begin{cases} I_p \times \text{background} & \text{mod}_{\text{RGB}} - \text{th} < I_{\text{mod}_{\text{RGB}}}(x, y) < \text{mod}_{\text{RGB}} + \text{th} \\ I_p \times \text{foreground} & \text{otherwise} \end{cases} \dots\dots\dots(8)$$

Pada rumus (8), *th* adalah nilai *threshold*, $I(\text{mod}_{\text{RGB}}(x, y))$ adalah intensitas dari mod_{RGB} dari piksel (x, y) . Setelah semua piksel diproses, maka dibuatlah sebuah *binary image*. Piksel *foreground* akan diberi warna putih, sedangkan piksel *background* akan diberi warna hitam.

2.4 Android

Android adalah sistem operasi untuk telepon seluler berbasis *linux*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasinya sendiri untuk digunakan oleh bermacam-macam piranti bergerak. Android adalah sistem operasi smartphone layar sentuh seperti *iOS iPhone* dan *OS BlackBerry*. Android dikembangkan *Google* dan pertama kali muncul tahun 2007 dengan ponsel pertamanya *GT-Mobile*. *Google* merilis Android sebagai *OS open source* di bawah naungan *Open Handset Alliance*. Android adalah *OS* yang sangat baik, cepat dan kuat serta memiliki antar muka pengguna intuitif yang dikemas dengan pilihan dan fleksibilitas. Android terintegrasi dengan layanan *Google*

seperti *Gmail*, *Google Calendar*, *Google Contacts*, dan *Google Voice*, dan sempurna bagi pengguna yang menggunakan *google* (Salbino, 2014). Perkembangan Android sangat cepat. Di awal tahun 2012 saja ada 200 juta pengguna aktif Android, dan *google play* mampu menampung 400.000 aplikasi yang siap digunakan, dan total mencapai 10 triliun kali aplikasi yang sudah didownload lewat *Android Market*. Jumlah ini diyakini akan terus bertambah seiring waktu dan perkembangan teknologi (Wahadyo, 2013).

2.4.1 Sejarah Android

Pada bulan oktober 2003 *Android Inc* didirikan di Palo Alto, Calofornia oleh Andy Rubin (pendiri *danger*), Rich Miner (pendiri *wildfire Communications, Inc*), Nick Sears (Mantan *VP T-Mobile*), dan Chris White (Kepala Desain dan Pengembangan Antarmuka *Web TV*) mengembangkan *Smartphone* yang lebih sadar akan lokasi dan preferensi penggunanya. Awal tujuan pengembangan Android yaitu mengembangkan sebuah sistem operasi canggih yang ditujukan untuk kamera digital, namun pasar untuk perangkat kamera digital tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar *Smartphone* untuk menyaingi *Symbian* dan *Windows Mobile*. Pada tanggal 5 November 2007, *Open Handset Alliance (OHA)* didirikan. *OHA* bertujuan mengembangkan standar terbuka bagi perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya. Sebuah platform perangkat seluler yang menggunakan kernel *Linux* versi 2,6 (Salbino, 2014).

2.4.2 Keunggulan dan Kelemahan Android

Salbino (2014) telah memaparkan penjelasan mengenai keunggulan dan kelemahan Android adalah sebagai berikut :

a. Keunggulan Android

- 1) Android bersifat *open source* karena berbasis *linux* yang dapat dikembangkan.
- 2) Akses mudah ke *Android App Market* untuk mendownload aplikasi secara gratis.
- 3) Sistem operasi dengan mudah memberitahukan notifikasi mengenai adanya *SMS*, *Email*, maupun artikel baru dari *RSS Reader*, bahkan misscall tidak terlewatkan sekalipun.
- 4) Sistem operasinya mendukung semua layanan *google*.
- 5) Instal *ROM* yang dapat dimodifikasi.

b. Kelemahan Android

- 1) Selalu memerlukan koneksi internet yang aktif. Minimal adanya koneksi internet *GPRS*.
- 2) Versi resmi Android kadang lambat dikeluarkan perusahaan perangkat.
- 3) Kadang masih terdapat malware karena *Android market* kurang kontrol dari pengelola.

2.5 Recall and Precision

Pengukuran efektivitas suatu sistem temu kembali dapat dilakukan dengan perhitungan terhadap nilai perolehan (*recall*), nilai ketepatan (*precision*), dan jatuhnya semu (*fallout*). Namun, diantara metode tersebut, perhitungan ketepatan merupakan cara yang paling umum digunakan (Tague-Sutcliffe, 1996).

Recall adalah proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian di sistem IR. Rumusnya adalah jumlah dokumen relevan yang ditemukan / jumlah semua dokumen relevan di dalam koleksi. Lalu, *precision* adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan si pencari informasi. Rumusnya: Jumlah dokumen relevan yang ditemukan / Jumlah semua dokumen yang ditemukan (Lestari, 2016).

Sedangkan *precision* dapat diartikan sebagai kepersisan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu). Jika seseorang mencari informasi di sebuah sistem, dan sistem menawarkan beberapa dokumen, maka kepersisan ini sebenarnya juga adalah relevansi. Artinya, seberapa persis atau cocok dokumen tersebut untuk keperluan pencari informasi, bergantung pada seberapa relevan dokumen tersebut bagi si pencari (Lestari, 2016).

Lancaster (1991) merumuskan matriks terkenal berikut ini sebagai ukuran *recall-precision*.

Tabel 2.3 Matriks *Recall and Precision*

	Relevan	Tidak Relevan	Total
Ditemukan	a (hits)	b (noise)	a + b
Tidak Ditemukan	c (misses)	d (rejected)	c + d
Total	a + c	b + d	a + b + c + d

Berdasarkan tabel tersebut rumus *recall-precision* menjadi sebagai berikut.

$$Recall = [a / (a+c)] \times 100.....(9);$$

$$Precision = [a / (a+b)] \times 100.....(10)$$

Melalui rumus ini, dapat dibayangkan bahwa sebuah sistem harus meningkatkan nilai *recall* dengan memperbesar nilai a. Nilai a yang besar ini dapat

terjadi jika jumlah dokumen yang diberikan sebuah sistem dalam sebuah pencarian juga besar. Semakin besar jumlah dokumen yang diberikan, semakin besar kemungkinan nilai *a*. Tetapi pada saat yang sama, muncul kemungkinan bahwa nilai *b* (atau jumlah dokumen yang tidak relevan) juga semakin besar. Ini artinya, nilai *precision*-nya semakin kecil. Dalam berbagai eksperimen, ditemukan kenyataan bahwa nilai *recall* dan *precision* ini cenderung berbanding terbalik. Jika *recall* tinggi, besar kemungkinan *precision* rendah (Lestari, 2016).

Kedua ukuran di atas biasanya diberi nilai dalam bentuk persentase, 1 sampai 100 persen. Sebuah sistem informasi akan dianggap baik jika tinggalkan *recall* maupun *precision*-nya tinggi. Jika seseorang mencari dokumen tentang 'perpustakaan' dan sistem tersebut memiliki 100 buku tentang perpustakaan, maka kinerja yang paling baik adalah jika sistem tersebut berhasil menemukan 100 dokumen tentang perpustakaan (Lestari, 2016).

Kalau sistem tersebut memberikan 100 temuan, dan pada temuan tersebut mendapat 50 dokumen tentang perpustakaan, maka nilai *recall* adalah 50% dan nilai *precision* adalah 50% juga. Kalau sistem tersebut memberikan 1 dokumen saja, dan dokumen tersebut relevan, maka *recall* akan bernilai 1% tetapi *precision* akan bernilai 100%.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A