

BAB II

LANDASAN TEORI

2.1 Fermentasi *Dough*

Dough berbahan dasar ragi memiliki proses yang paling penting dan paling mendasar yaitu proses fermentasi, dimana ragi berperan sebagai komponen utama yang mengembangkan, mematangkan, memproduksi senyawa-senyawa gas dan aroma dari *dough* (Dafen, 2011).

Tahap awal dalam proses fermentasi merupakan *resting* atau *proofing* roti yaitu roti didiamkan dan membiarkan ragi bekerja. Ragi bekerja dengan mengubah gula dalam adonan menjadi gas karbondioksida (CO₂), untuk mematangkan dan melunakkan gluten di dalam adonan. Gluten yang sudah matang dan lunak akan mengembangkan gas secara merata dan menahannya yang akan membentuk gelembung-gelembung pada adonan yang juga membentuk cita rasa, struktur berbentuk *porous* seperti busa, tekstur yang lembut, dan membentuk pori untuk perkembangan *dough*. Suhu ideal untuk proses fermentasi adalah 35° celsius dan kelembaban udara 75% (Mudjajanto, 2004).

Waktu yang dibutuhkan pada saat *resting* tidak dapat ditetapkan karena kelembaban udara dan suhu tidaklah selalu sama setiap tempatnya, jika cuaca terlalu dingin atau panas, proses fermentasi dari ragi akan lambat bekerja bahkan dapat menyebabkan ragi mati. Pada proses fermentasi yang baik adonan akan mengembang dua kali lipat. Dalam tahap fermentasi juga dapat terjadi kesalahan yaitu *overproofing* atau *underproofed*. *Overproofing* adalah kondisi ketika *dough resting* terlalu lama pada saat fermentasi. Hal ini menyebabkan gelembung dalam roti menjadi sangat besar sampai menjadi lubang. Sedangkan *Underproofed*

merupakan kondisi ketika waktu *dough resting* kurang lama sehingga proses fermentasi belum sempurna (Bernstein, 2014).

Pada umumnya untuk melakukan pengecekan pada proses fermentasi dengan cara menekan adonan dan melihat apakah adonan tersebut kembali ke bentuk semula atau tidak. Jika adonan tidak kembali ke bentuk semula dengan cepat berarti sudah baik, sebaliknya jika langsung kembali ke bentuk semula berarti masih membutuhkan waktu untuk *dough resting* (Bernstein, 2014).

2.2 Image Processing

Image Processing merupakan pengolahan terhadap sebuah gambar dengan berbagai tujuan seperti peningkatan kualitas, pemulihan gambar, mengambil informasi dari gambar, mengubah dimensi atau ukuran gambar (Girod, 2015).

Terdapat dua jenis metode yang digunakan untuk *image processing* yaitu *analogue* dan *digital image processing*. *Analogue image processing* digunakan untuk *hard copies* seperti cetakan dan foto-foto. Teknik *Digital image processing* membantu dalam memanipulasi gambar *digital* menggunakan computer (Anbarjafari, 2014).

2.3 Bitmap Image

Bitmap Image dikenalkan oleh Microsoft sebagai format data gambar dasar antara *users* dengan sistem operasi Windows. Sekarang format data *bitmap* sudah didukung banyak sistem operasi dan *file systems*, tetapi semakin lama semakin dikit yang menggunakan format ini, karena memiliki ukuran data yang besar (Elgabar, 2013).

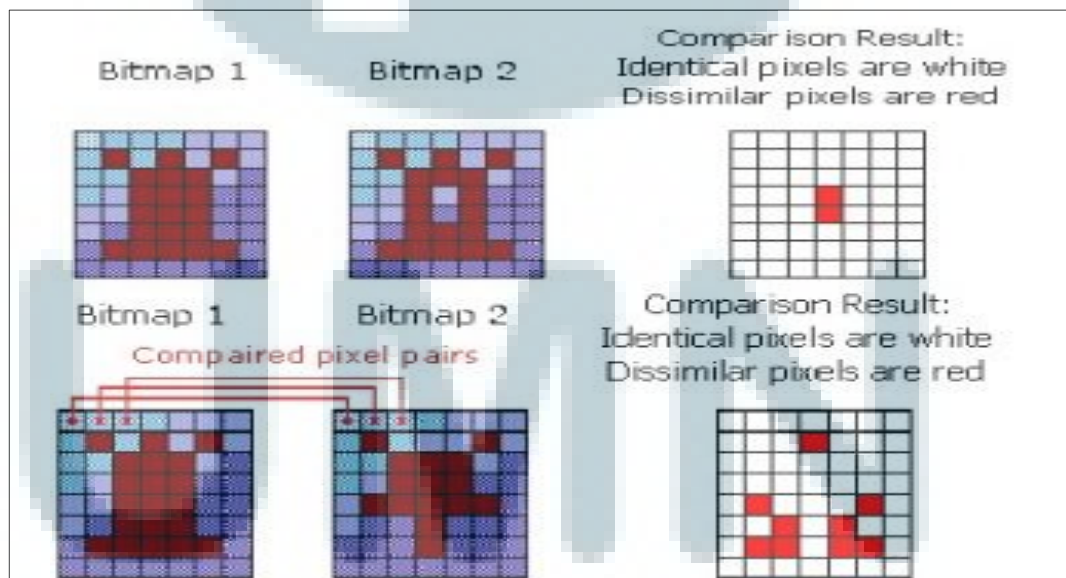
Bitmap image merupakan gambar dua dimensi yang memiliki domain persegi dan memiliki dua koordinat yaitu x dan y yang berguna untuk menentukan

titik-titik dalam gambar dan menyimpan *binary image* yaitu warna yang ada pada titik koordinat tersebut (Foley, 1996). Perbedaan JPEG atau JPG dengan BMP hanya pada kompresinya saja, tetapi memiliki cara penyimpanan *binary image* yang sama.

2.4 Bitmap Image Comparison

Image Comparison merupakan salah satu proses penting dalam *image processing*. Terkadang perlu untuk membandingkan dua gambar untuk memperkirakan kesamaan dan ketidaksamaan di antara keduanya. Estimasi dari kesamaan gambar merupakan masalah penting dalam *image analysis*. Mengukur kesamaan diantara kedua gambar berguna untuk perbandingan algoritma yang ditujukan untuk *noise reduction*, *image matching*, *image coding*, dan *restoration*.

Perbandingan gambar dapat dilakukan dengan *pixel by pixel* yaitu membandingkan warna *pixel* di koordinat yang sama di dalam kedua gambar, namun dibutuhkan kedua gambar yang identik.



Gambar 2.1 Pengecekan Perbedaan dan Persamaan *Pixel* (Katukam, 2015).

Menentukan perbedaan warna yang dianggap identik dari kedua *pixel*. Perbedaan warna direpresentasikan sebagai nilai *integer* antara 0-255 yang menentukan perbedaan yang dapat diterima untuk setiap komponen warna (*red*, *green*, dan *blue*) dari *pixel* yang dibandingkan. Kedua *pixel* dikatakan identik jika perbedaan antara kedua komponen warna tidak melebihi nilai yang ditentukan. Ketika toleransi warna adalah 0, yaitu nilai *default*, *pixel* yang dibandingkan harus memiliki warna yang persis sama. Ketika toleransi warna adalah 255, warna di setiap *pixel* akan dianggap identik (Katukam, 2015).

Image comparison sering dilakukan dengan menghitung *correlation function*, *the root of the mean square-error* atau *measures of the signal-to-noise ratio*. Pendekatan terakhir hanya berlaku jika mempunyai pengetahuan yang cukup tentang konten gambar. Dalam kasus gambar biner, masalah perbandingan gambar menjadi lebih sederhana karena konteks pada gambar gampang diambil dengan asumsi *pixel* hitam sebagai objek dan *pixel* lainnya sebagai *background* (Gesu, 1999).

2.5 Naïve Bayes

Naïve bayes merupakan salah satu algoritma yang efisien untuk *machine learning* dan *data mining*. Naïve bayes sangat kompetitif dalam proses klasifikasi dengan menggunakan asumsi atribut independen. Menurut Pattekari dan Parveen (2012), keuntungan menggunakan Naïve Bayes yaitu hanya membutuhkan data *training* dalam jumlah sedikit untuk menentukan klasifikasinya. Naïve bayes merupakan teorema dasar Bayes yang ditambahkan asumsi independen Naïve. Pada teorema Naive Bayes memiliki rumus seperti (2.1) (Natalius, 2010).

$$P(y | x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad \dots(2.1)$$

Dari rumus (2.1) $P(y \mid x_1, \dots, x_n)$ merupakan *posterior* yaitu peluang munculnya kelas y dengan karakteristik dari x_1 sampai x_n . $P(y)$ merupakan *prior* yaitu peluang munculnya kelas y . $P(x_1, \dots, x_n \mid y)$ merupakan *likelihood*, dan $P(x_1, \dots, x_n)$ merupakan *evidence*. (Raschka, 2014).

$$P(x_1, \dots, x_n \mid y) = P(x_1 \mid y) P(x_2 \mid y) \dots P(x_n \mid y) \quad \dots(2.2)$$

Rumus (2.2) adalah cara perhitungan *likelihood* yaitu semua peluang muncul kelas y dengan karakteristik dari x_1 sampai dengan x_n dikalikan.

$$P(x_i) = P(x_i \mid y_j) \cdot P(y_j) + P(x_i \mid y_j^n) \cdot P(y_j^n) \quad \dots(2.3)$$

Rumus (2.3) adalah cara perhitungan *evidence* dengan (y_j^n sebagai “complement” dan pada dasarnya diterjemahkan menjadi “not class y_j .”) (Raschka, 2014).

UMMN