

# Klasifikasi gambar makanan jepang menggunakan pendekatan KAZE dan Machine Learning

Singgih Putra P<sup>1</sup>, Roby Sirojul A<sup>2</sup>, Ilham Ade W.S.<sup>3</sup>, Rayhan Aulia<sup>4</sup>

<sup>1,2,3,4</sup> (Program Studi Teknik Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur)

<sup>1</sup>[singgihputrap.if@gmail.com](mailto:singgihputrap.if@gmail.com)

<sup>2</sup>[Roby.sirojul@gmail.com](mailto:Roby.sirojul@gmail.com)

<sup>3</sup>[ilhamade.if@gmail.com](mailto:ilhamade.if@gmail.com)

<sup>4</sup>[rayhanaulia.if@gmail.com](mailto:rayhanaulia.if@gmail.com)

**Abstrak**— Belakangan ini gambar menjadi salah satu bidang penelitian yang populer. Ilmu komputer berusaha untuk mempelajari dan mengenali berbagai macam objek gambar. Dalam penelitian ini adalah klasifikasi gambar makanan jepang menggunakan KAZE dengan tujuan mengetahui seberapa efektif metode KAZE dalam hal ekstraksi fitur guna menunjang klasifikasi makanan jepang. Dalam jurnal ini akan menggunakan KAZE dalam hal ekstraksi fitur gambar, lalu menggunakan K-means dalam klusterting dan menggunakan KD-tree untuk image recognition. Nantinya percobaan akan menggunakan tiga tipe makanan Jepang yaitu nasi (rice), belut diatas nasi (eels on rice), Pilaf, dan Chicken-'n'-egg on rice. Dari hasil percobaan menunjukkan bahwa pengenalan objek gambar makanan Jepang menggunakan algoritma KAZE menunjukkan hasil yang cukup baik dengan tingkat akurasi sebesar 43,6%.

**Kata Kunci**— Object Recognition, keypoint, KAZE, KD-Tree, Food Image.

## I. PENDAHULUAN

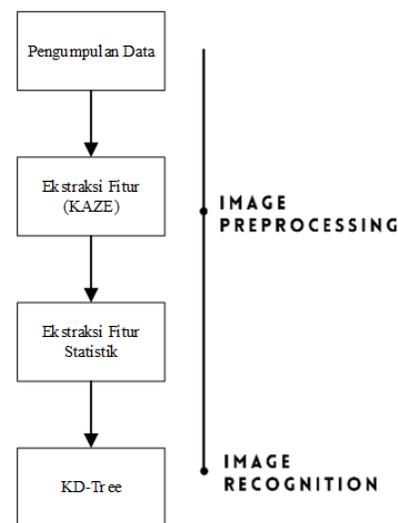
Industri makananan adalah industri yang tidak ada matinya. Banyak sekali tren-tren global seputar makanan. Oleh karenanya banyak sekali penelitian yang bergerak pada bidang makanan, tidak terkecuali komputer sains. Komputer mencoba berbagai objek makanan untuk dikenali. Penelitian ini mulai dikembangkan semenjak orang-orang terbiasa untuk mengambil gambar makanan di setiap kegiatannya. Mengambil gambar seperti itu menciptakan peluang untuk mengekstrak informasi dari makanan tersebut yang didalamnya terdapat informasi kesehatan, kandungan nutrisi, dll. Pengenalan makanan (food recognition) mulai populer karena hal tersebut.

Dalam beberapa penelitian sudah didapat dataset dari berbagai macam makanan dari berbagai negara, diantaranya adalah makanan barat dan makanan jepang[1][2]. Dalam penelitian ini kami menggunakan dataset makanan jepang dengan berbagai teknik ekstraksi informasi untuk mendapatkan hasil yang diinginkan. Untuk itu gambar yang jelas dan presentasi makanan sangat berpengaruh pada hasil dari penelitian ini. Kami menggunakan beberapa jenis makanan diantaranya yaitu nasi (rice), belut diatas nasi (eels on rice), Pilaf, dan Chicken-'n'-egg on rice.

## II. METODOLOGI

Penelitian ini di bagi menjadi 3 langkah, tahap pertama data collection, tahap kedua image pre-processing, dan tahap ketiga image recognition. Image pre-processing

menggunakan ekstraksi fitur KAZE, fitur kluster menggunakan K-MEANS, dan transformasi fitur untuk mendapatkan fitur statistik. Skema proses penelitian diberikan pada Gambar 1 detail masing-masing metode disajikan pada bagian berikutnya.



Gbr. 1 Diagram alir penelitian.

### A. Pengumpulan Data

Penelitian ini menggunakan makanan Jepang sebagai dataset, gambar di ambil dari (<http://foodcam.mobi/dataset100.zip>). Lalu gambar diseleksi untuk menentukan apakah gambar yang di ambil merupakan gambar yang benar atau tidak. Gambar harus memenuhi kriteria yang ditentukan, karena apabila tidak memenuhi maka akan memberngaruhi hasil dan keefektifan dari penelitian yang ada. Gambar yang diambil sejumlah 100 gambar yang terdiri dari 4 jenis. Berikut ini kriteria gambar yang digunakan dalam penelitian :

1. Gambar harus makanan jepang
2. Gambar utuh bukan berupa potongan
3. Gambar makanan tidak banyak tercampur dengan fokus objek lain
4. Satu gambar mewakili satu jenis makanan



Gbr. 2 Contoh gambar makanan yang digunakan.

Keterangan gambar : di urutkan dari kiri atas sampai kanan bawah

1. Rice
2. Eels on rice
3. Pilaf
4. Chicken-'n'-egg on rice

## B. Image Preprocessing

### 1) KAZE:

KAZE adalah algoritma yang digunakan untuk deteksi fitur. Gagasan di balik pembuatan algoritma ini adalah untuk mendeteksi dan menggambarkan fitur 2D dalam ekstrem ruang skala nonlinear untuk mendapatkan akurasi lokalisasi dan keunikan/ kekhasan. Gaussian blurring digunakan dalam algoritma pengenalan objek lainnya, seperti SIFT, tidak memandang batas objek dimana detail gambar dan noise dihaluskan ke tingkat yang sama di semua level skala. Dalam pengenalan objek, KAZE utamanya memiliki langkah ekstraksi yang sama dengan SIFT namun dengan beberapa perbedaan dalam tiap stepnya. Berikut ini adalah step dari KAZE[5]:

#### a. Menghitung ruang skala nonlinear

Kami mengambil pendekatan yang sama seperti yang dilakukan dalam SIFT, mendiskritisasi ruang skala dalam logaritmik langkah-langkah diatur dalam serangkaian sub-tingkat  $O$  oktaf dan  $S$ . Perhatikan bahwa kami selalu bekerja dengan resolusi gambar asli, tanpa melakukan downsampling di setiap baru oktaf seperti yang dilakukan dalam SIFT. Himpunan oktaf dan sub-level diidentifikasi oleh diskrit indeks oktaf  $o$  dan sub-level satu  $s$ . Oktaf dan indeks sub-level dipetakan untuk skala yang sesuai mereka  $\sigma$  melalui rumus berikut:

$$\sigma_i(o, s) = \sigma_0 2^{o+s/S}, \quad o \in [0 \dots O-1], \quad s \in [0 \dots S-1], \quad i \in [0 \dots N] \quad (1)$$

#### b. Deteksi fitur

Untuk mendeteksi *Point of Interest*, Kami menghitung respons penentu skala-ternormalisasi hessian pada berbagai tingkat skala untuk mendeteksi *Point of Interest*. Matriks Hessian dihitung menggunakan:

$$\text{Error! Reference source not found.} \quad (2)$$

di mana  $L_{xx}$ ,  $L_{yy}$  memiliki turunan horisontal dan vertikal orde dua secara berurutan, dan  $L_{xy}$  adalah turunan silang urutan kedua. Kami menggunakan *Nearest Neighbour* berjumlah 5.

#### c. Membuat Deskriptor

KAZE menghasilkan deskriptor untuk setiap keypoint yang merupakan skala dan invarian rotasi. Penjelasan untuk setiap keypoint dibuat oleh mengambil sampel  $16 \times 16$  piksel di sekitar keypoint dan membaginya menjadi  $4 \times 4$  blok, kemudian menghitung histogram dengan 8 bins untuk setiap blok dan deskriptor akan menjadi 128 dimensi vector[3].

### 2) Feature Transformation:

Transformasi Fitur menggunakan fitur statistika setelah menggunakan KAZE. Fitur statistik digenerasi dari setiap kluster yang ada dengan total 7 kluster. Fitur statistik yang digunakan berjumlah 5 yaitu Mean, Max, Min, Kurtosis, dan Skewness yang dapat dihitung menggunakan rumus dibawah ini:

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^N x_i, \quad (3)$$

$$x_{\max} = \max(x_j), \quad (4)$$

$$x_{\min} = \min(x_j), \quad (5)$$

$$\text{skewness}_j = \frac{\sum_{i=1}^N (x_i - \bar{x}_j)^3}{N\sigma_j^3}, \quad (6)$$

$$\text{kurtosis}_j = \frac{\sum_{i=1}^N (x_i - \bar{x}_j)^4}{N\sigma_j^4}, \quad (7)$$

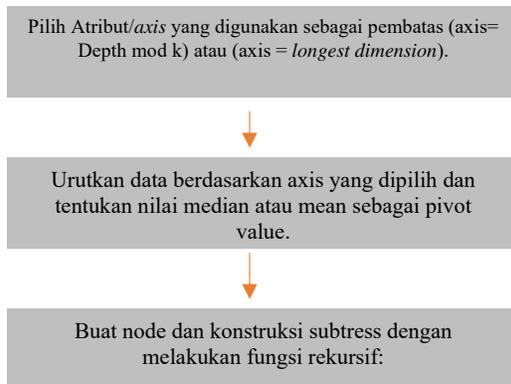
## C. Image Recognition

### 1) K-Dimensional Tree (KD-Tree)

adalah data struktur yang bersifat *space-partitioning* dan merupakan kasus spesial dari *binary space partitioning tree*. KD-Tree bertujuan untuk mengatur poin-poin dalam  $k$ -dimensional space. KD-Tree umumnya diaplikasikan dalam pencarian yang memiliki banyak kunci pencarian (*multidimensional key*) seperti *range search*, dan *nearest neighbor search*[4].

Dalam implementasi, KD-Tree adalah binary tree dimana setiap node pada binary tree tersebut adalah sebuah poin berdimensi  $k$ . Setiap node yang bukan merupakan leaf pada KD-Tree akan menghasilkan sebuah hyperplane yang memisahkan sebuah ruang menjadi 2 bagian. Setiap poin yang berada di daerah sebelah kiri hyperplane merepresentasikan node yang berada di subtree sebelah kiri. Demikian pula dengan setiap poin yang berada di daerah sebelah kanan

hyperplane akan merepresentasikan node yang berada di subtree sebelah kanan.  
 Alur metode dari fungsi pembentukan KD-Tree dapat dilihat pada Gambar 1.3



Gbr. 4 Diagram alir KD-Tree.

Keterangan :

- a.  $node.leftChild := kdtree(poin\ sebelum\ pivot\ value, depth+1)$
- b.  $node.rightChild := kdtree(poin\ setelah\ pivot\ value, depth+1)$

Untuk menentukan dimensi yang menjadi pemisah pada Kdimensional Tree terdapat 2 metode.

1. *Menggunakan modulo dari nilai kedalaman node*

Pada metode ini semua node pada kedalaman yang sama akan memiliki dimensi pemisah yang sama.

2. *Rentang dimensi*

Pada setiap node akan dicari dimensi yang memiliki rentang terpanjang. Dimensi dengan rentang terpanjang tersebut akan digunakan sebagai dimensi pemisah pada node tersebut.

III. HASIL DAN ANALISA

Ada lima KD-Tree yang digunakan dalam penelitian ini, setiap KD-Tree dibangun dari setiap fitur statistik (masing-masing statistik fitur berisi 7 atribut). KD-Tree pertama dibangun dari atribut kurtosis, yang kedua dibangun dari kemiringan atribut, yang ketiga dibangun dari atribut rata-rata, yang keempat dibangun dari atribut max, dan yang terakhir satu dibangun dari atribut min. Lima KD-Tree berbeda digunakan karena hanya sekitar 100 data yang digunakan dalam hal ini penelitian, sehingga kedalaman pohon hanya mencapai 5 yang berarti jika satu KD-Tree digunakan, hanya 5 atribut yang dipertimbangkan dalam metode ini. Dalam langkah pengenalan dengan KD-Tree, 7 atribut kurtosis dari gambar diteruskan ke KD-Tree pertama, 7 atribut kemiringan gambar diteruskan ke KD-Tree kedua, 7 atribut rata-rata gambar dilewatkan ke KD-Tree ketiga, 7 maks atribut gambar adalah diteruskan ke KD-Tree keempat, 7 menit atribut gambar dilewatkan ke KD-Tree kelima. Tujuan dari langkah pengenalan adalah untuk mengklasifikasikan gambar yang tidak berlabel berdasarkan pada model. Setiap k-d tree memilih k label untuk gambar berdasarkan atribut yang

dilewati, label dari setiap KD-Tree digabungkan dan akan digunakan untuk menemukan label yang sering muncul (memiliki suara tertinggi dari gabungan tersebut suara) untuk menjadi label gambar. Nilai k yang berbeda diterapkan untuk percobaan ini.

TABEL I  
 HASIL AKURASI TRANFORMASI

No	Feature Transformation	Akurasi
1	Mean	32,78%
2	Min	29,52%
3	Max	36,35%
4	Kurtosis	42,5%
5	Skewness	45,2%

IV. KESIMPULAN

Makalah ini mengusulkan KAZE dan pendekatan pembelajaran mesin K-D Tree dalam pengenalan gambar makanan untuk empat jenis makanan Jepang, yaitu nasi (*rice*), belut diatas nasi (*eels on rice*), *Pilaf*, dan *Chicken-n'-egg on rice*. Hasil eksperimen menunjukkan hasil KAZE menghasilkan fitur dimensi tinggi yang harus dikurangi untuk menangani kompleksitas pemrosesan. Fitur statistik penting di sini, untuk setiap cluster memiliki nilai pusat yang serupa. KD-Tree terbukti memiliki akurasi dalam pengenalan gambar makanan. Hasilnya mencapai 43,6%. Akurasi ini relatif rendah sejak itu variasi tampilan gambar cukup tinggi untuk setiap jenis makanan. Diperlukan data yang besar untuk mendapatkan akurasi gambar yang lebih tinggi terhadap pengenalan makanan terutama untuk Makanan Jepang.

REFERENSI

- [1] Giovany, Stanley. 2017. Machine Learning and SIFT Approach for Indonesian Food Image Recognition: Procedia Computer Science: vol.116. 612–620.
- [2] Ze-Ping Cai, De-Gui Xiao. 2014. Feature matching algorithm based on KAZE and fast approximate nearest neighbor search.
- [3] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Intl. J. of Computer Vision 60 (2004) 91–110
- [4] Zhao lulu, Geng Guohua, Li Kang, He A-Jing, "Image matching algorithm based on SURF and fast approximate nearest neighbor search," Computer application research, 2012
- [5] Shaharyar Ahmed Khan Tareen, Zahra Saleem, " A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK, " International Conference on Computing, Mathematics and Engineering Technologies, 2018