

## ANALISIS PERFORMANSI *IMAGE RETRIEVAL* BERDASARKAN *COLOR FEATURE* DAN *SCALE INVARIANT FEATURE TRANSFORM*

### PERFORMANCE ANALYSIS *IMAGE RETRIEVAL* BASED ON *COLOR FEATURE* AND *SCALE INVARIANT FEATURE*

Viky Premeita Mitayani<sup>1</sup>, Suryo Adhi Wibowo<sup>2</sup>, Rissa Rahmania<sup>3</sup>

<sup>1,2</sup>. Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

<sup>1</sup>[vikypremeita@student.telkomuniversity.ac.id](mailto:vikypremeita@student.telkomuniversity.ac.id), <sup>2</sup>[suryoadhiwibowo@telkomuniversity.ac.id](mailto:suryoadhiwibowo@telkomuniversity.ac.id)

<sup>3</sup>[saniarahmani@telkomuniversity.ac.id](mailto:saniarahmani@telkomuniversity.ac.id)

#### Abstrak

Berkembangnya kemajuan teknologi berpengaruh kepada gaya hidup yang terjadi pada masyarakat, contohnya sosial media yang banyak menggunakan citra sebagai objeknya. Karena banyaknya citra yang ada, sulit untuk mencari citra yang ingin ditemukan, dengan begitu *Image Retrieval* terbentuk sebagai teknik pengambilan citra dengan skala yang besar. Dalam kehidupan sehari-hari *Image Retrieval* sudah banyak dijumpai contohnya adalah *Google Images* yang berfungsi sebagai mesin pencari dengan menggunakan data citra. Berdasarkan permasalahan tersebut, dalam Tugas Akhir ini penulis merancang sistem dengan menggunakan metode *Color Feature* dan *Scale Invariant Feature Transform* untuk dapat menemukan data citra yang dicari. *Color Feature* merupakan metode yang digunakan untuk mengidentifikasi warna. *Scale Invariant Feature Transform* merupakan fitur untuk mendeteksi algoritma yang terdapat dalam citra. Namun penulis tidak menggunakan metode tersebut secara terpisah, metode yang digunakan dalam Tugas Akhir ini menggabungkan dua metode tersebut untuk mengoptimalkan pengambilan citra. Hasil yang diharapkan dalam penelitian ini adalah peningkatan akurasi dalam pengambilan data citra dalam skala yang besar, dengan harapan mendapat nilai *Korelasi* yang tinggi sehingga penelitian ini dapat dimanfaatkan untuk penelitian selanjutnya di waktu yang akan datang.

**Kata Kunci :** *image retrieval, scale invariant feature transform, color feature.*

#### Abstract

The development of technological advances affects the lifestyle that occurs in society, for example, social media that uses a lot of images as objects. Because of the many existing images, it is difficult to find the image you want to find, so *Image Retrieval* is formed as a large-scale image retrieval technique. In everyday life *Image Retrieval* has often been found for example is *Google Images* which functions as a search engine using image data. Based on these problems, in this Final Project the author designs a system using the *Color Feature* and *Scale Invariant Feature Transform* methods to be able to find the image data sought. *Color Feature* is a method used to identify colors. *Scale Invariant Feature Transform* is a feature to detect the algorithm contained in the image. But the authors do not use these methods separately, the method used in this Final Project combines the two methods to optimize image capture. The expected result in this research is an increase in accuracy in image capture data on a large scale, with the hope of getting a high *Korelasi* so that this study can be utilized for further research in the future.

**Keywords:** *image retrieval, scale invariant feature transform, color feature.*

#### 1. Pendahuluan

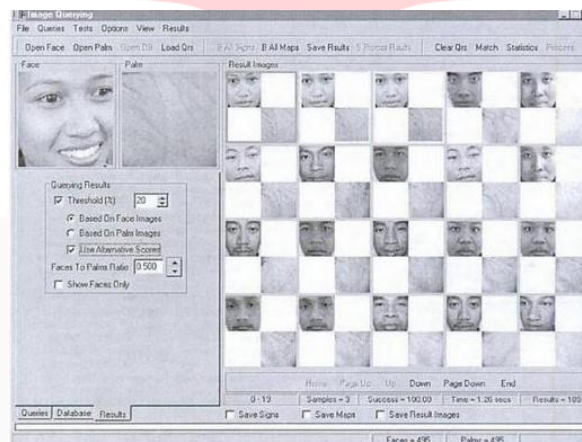
Seiring dengan perkembangan teknologi, banyak citra yang tersimpan di dalam suatu bentuk data dan kemudian disimpan ke media penyimpanannya. Data citra yang berlebihan menyebabkan sulit untuk pencarian data citra yang dimaksud, maka *Image Retrieval* terbentuk sebagai teknik untuk pengambilan gambar tersebut. *Image retrieval* di dalam kehidupan sehari-hari pun sangat bermanfaat, contohnya adalah *Google Images* yang merupakan mesin pencarian dengan menggunakan data citra. Pada tahun 1999 David Lowe mempublikasikan metode baru dalam *Computer Vision* bernama *Scale Invariant Feature Transform* (SIFT) untuk mendeteksi dan mendeskripsikan fitur-fitur lokal pada citra itu sendiri [1]. Fitur SIFT bekerja dengan nilai *invariant* yang dibesarkan ke skala, iluminasi dan lainnya. Setelah itu tahun 2011 penelitian yang berjudul Analisis dan Implementasi *Scale invariant feature Transform* (SIFT) pada *Content Based Image Retrieval* yang ditulis oleh Ivan Dwi Putra dijelaskan bahwa, sistem *image retrieval* menggunakan SIFT merupakan sistem yang tahan terhadap perubahan modifikasi skala pada citra. Terdapat hubungan antara skala yang digunakan sebagai *query*

terhadap hasil *retrieval* dimana ukuran skala yang di *retrieve* akan mengikuti ukuran skala *query* [2]. Namun pada penelitian sebelumnya belum terdapat *indexing* serta di dalam *Bag-of-Words* (BoW) *power* dari SIFT dinilai masih kurang, tahun 2014 dilakukan perpaduan fitur pada *indexing level* yang disebut dengan *coupled Multi-Index* (c-MI). C-MI bertujuan untuk meningkatkan keakuratan dalam pengambilan data citra agar dapat menemukan data citra yang mirip dalam *real-time* [3]. Karena banyaknya kemiripan (*nearest neighbor*) dalam *dataset* yang begitu besar dan mengakibatkan banyak kemunculan perhatian dari berbagai kalangan menjadikan ini sebagai masalah *research* yang besar. Untuk mengatasi masalah tersebut algoritma dari *Approximation Nearest Neighbor* (ANN) diteliti lebih lanjut berdasarkan faktor *approximation*-nya, tempat dan juga waktu *query*-nya [4].

## 2. Dasar Teori

### A. Image Retrieval

*Image Retrieval* merupakan aplikasi pengolahan citra yang dapat membantu pengguna mengambil atau mencari dengan cepat suatu citra pada suatu database citra berdasarkan *query* atau permintaan pengguna. *Database* pada umumnya memiliki ukuran dalam skala besar. Gambar 2.1 berikut menunjukkan contoh aplikasi *image querying* untuk citra wajah dan telapak tangan kemudian sistem akan mencari dan menampilkan citra-citra yang mirip dengan *query* [5].



Gambar 2.1 Aplikasi *image querying* citra wajah dan telapak tangan [5]

### B. Scale Invariant Feature Transform (SIFT).

SIFT merupakan salah satu fitur yang digunakan dalam c-MI, di mana merupakan fitur yang mendeteksi algoritma yang ada di *computer vision*, bertujuan untuk mendeteksi fitur lokal yang ada pada gambar. SIFT sendiri telah dipatenkan di Canada oleh David Lowe pada tahun 1999. Objek dari *keypoint* SIFT pertama diekstraksi dari beberapa citra dan disimpan di dalam *database* [1]. Kemudian objek *recognized* ke dalam citra yang baru setelah dibandingkan fiturnya satu per satu ke dalam *database* lalu citra yang serupa akan ditemukan dengan cara *Euclidean Distance*. Dimana *Euclidean Distance* merupakan cara untuk menghitung jarak dari 2 titik yang akan ditentukan. Tahapan yang terdapat pada algoritma ini adalah sebagai berikut:

#### 1. Mencari Nilai Ekstrim dalam Skala Ruang

Langkah pertama adalah melakukan pencarian nilai ekstrim yang terdapat pada skala ruang untuk menentukan *keypoint* yang terdapat pada suatu citra. Dengan fungsi *Gaussian*:

$$L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y) \quad (2.1)$$

dimana  $L(x,y,k\sigma)$  merupakan citra pada skala ruang,  $G(x,y,k\sigma)$  merupakan skala variabel *Gaussian* dan  $I(x,y)$  merupakan citra masukan. Setelah itu dilakukan tahap pencarian citra hasil *Difference-of-Gaussian*,  $D(x,y,\sigma)$ , yang didapat dengan meng-konvolusi citra masukan dengan filter *Difference-of-Gaussian*, maka dalam skala  $k$  dapat dilihat dalam persamaan:

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad (2.2)$$

Setelah itu citra hasil dari konvolusi di kelompokkan berdasarkan *octave* (satu *octave* setara dengan penggandaan nilai  $\sigma$ ), dan nilai  $k$  sudah ditetapkan sehingga diperoleh jumlah citra *blur* yang sama pada tiap *octave* dan didapat pula citra hasil *Difference-of-Gaussian* yang sama pada tiap *octave*.

#### 2. Menentukan Keypoint

Setelah didapat citra hasil *Difference-of-Gaussian* yang sama pada tiap *octave*, langkah selanjutnya yaitu mencari kandidat untuk *keypoint*. Kandidat *keypoint* didapat dengan mendeteksi titik maksimum lokal atau titik minimum lokal, untuk mencari nilai maksimum atau minimum lokal maka piksel yang terdapat pada citra dari hasil *Difference-of-Gaussian* dibandingkan dengan 8 piksel yang berada di sekitarnya. Jika setelah dibandingkan piksel tersebut merupakan nilai maksimum atau minimum lokal, maka selanjutnya piksel tersebut dijadikan kandidat dari *keypoint*.

Setelah kandidat *keypoint* didapat, langkah selanjutnya adalah mengurangi kandidat tersebut dengan cara mengambil detail skala, lokasi dan rasio kelengkungan inti dari kandidat *keypoint*. Jika kandidat *keypoint* memiliki nilai kontras yang rendah maka kandidat tersebut akan dihilangkan.

Dengan menggunakan persamaan deret Taylor pada fungsi skala ruang  $D(x, y, \sigma)$  yang digeser sehingga titik asal berpindah pada titik sampel:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.3)$$

dimana  $D$  dan penurunannya dilihat dari titik sampel, dan  $x = (x, y, \sigma)^T$  adalah offset dari titik ini

$$x = \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.4)$$

lalu fungsi diatas disubstitusikan untuk mendapat fungsi nilai ekstrim agar nilai ekstrim yang kurang stabil dapat dihilangkan, sehingga persamaannya dapat dilihat:

$$D(x) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} x \quad (2.5)$$

### 3. Penentuan Orientasi

Pada langkah ini *keypoint* yang didapat akan diberi suatu orientasi tetap berdasarkan sifat-sifat lokalnya pada citra, maka *keypoint* dapat direpresentasikan relatif terhadap orientasi sehingga *keypoint* tidak terpengaruh rotasi pada citra. Maka dari itu dilakukan perhitungan terhadap besarnya gradien dan juga arah orientasinya:

$$M(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.6)$$

$$\theta(x, y) = \frac{\tan^{-1}(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))} \quad (2.7)$$

dimana  $M(x, y)$  merupakan besarnya gradien dan  $\theta(x, y)$  merupakan arah orientasi.

### 4. Keypoint Descriptor

Untuk mendapatkan *keypoint* yang *invariant* terhadap perubahan intensitas cahaya maupun perubahan sudut pandang maka dilakukan tahap ini. Koordinat dari deskriptor dan juga gradien orientasi akan dirotasikan terhadap orientasi *keypoint*. *Keypoint Descriptor* yang sudah diperoleh akan dinormalisasi untuk mengatasi adanya pengaruh dari perubahan cahaya.

### C. Color Feature

Diantara fitur yang lain, warna secara visual merupakan yang sangat berpengaruh serta dapat diandalkan. Di dalam warna terdapat beberapa warna dasar di- antaranya ada 11 warna dasar, yaitu hitam, merah muda, orange, hijau, ungu, biru, abu-abu, putih, kuning, coklat dan merah [3]. Pada penelitian ini menerapkan Color Names (CN) sebagai *color feature*. Lalu CN menetapkan vektor 11-D untuk setiap piksel, dimana setiap entri mengkodekan salah satu dari sebelas warna dasar yang sudah dijelaskan sebelumnya. Dalam *computer vision*, atribut warna melibatkan penugasan label warna linguistik ke piksel pada suatu citra. CN *descriptor* didefinisikan sebagai vektor yang mengandung probabilitas nama warna yang diberikan pada wilayah  $R$ , yang dapat dijelaskan dengan persamaan (2.8)

$$CN = \{p(cn_1|R), \dots, p(cn_{11}|R)\} \quad (2.8)$$

### D. K-Nearest Neighbor

*K-Nearest Neighbor*(KNN) merupakan sebuah metode untuk melakukan klasifikasi pada objek berdasarkan pada data pembelajaran yang mempunyai jarak terdekat dengan objek tersebut. Algoritma ini mempunyai tujuan, yaitu untuk mengklarifikasikan objek baru berdasarkan data pembelajaran (data latihan), yang diambil dari nilai  $k$ , parameter  $k$  merupakan jumlah tetangga terdekat.

### E. K-Means

Teknik untuk *clustering* yang sering digunakan adalah algoritma *k-means* karena teknik ini termasuk mudah untuk diimplementasikan, dimana *k-means* merupakan proses permodelan tanpa supervisi. Algoritma *k-means* mempunyai 4 langkah, yaitu inialisasi *centroid*, mengklasifikasikan data, melakukan perhitungan titik pusat *cluster* baru dan yang terakhir adalah meningkatkan *clustering* sehingga nilai titik *centroid* tidak berubah. Namun *k-means* masih memiliki beberapa kekurangan, diantaranya adalah:

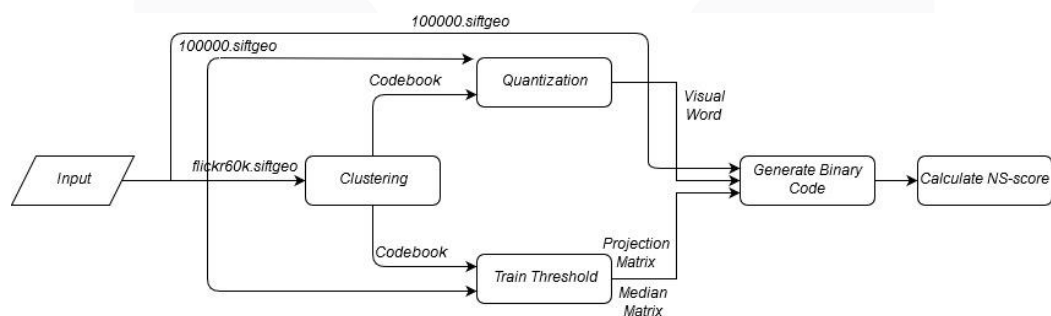
1. Masih menggunakan metode acak untuk pemilihan titik pusat *cluster*.
2. Membutuhkan waktu yang lama.
3. Jumlah *cluster* ditentukan secara manual oleh pengguna.

Karena *cluster k-means* menjadi sangat lambat maka dilakukan sebuah peningkatan yang bernama algoritma *Approximate K-Means (AKM)*, algoritma ini menggunakan pencarian *nearest neighbor* dengan menggunakan *k-d tree*[6], sehingga algoritma ini dua kali lebih cepat dibandingkan dengan algoritma *k-means*.

## 3. Perancangan

### A. Perancangan Sistem

Proses perancangan serta pengimplementasian pada bab ini dibutuhkan diagram blok yang dapat menggambarkan keseluruhan sistem dari *image retrieval* secara umum, metode yang digunakan dalam *image retrieval* adalah *coupled Multi-Index (c-MI)* yang merupakan penggabungan dari *Color Feature* dan SIFT, metode ini bekerja pada *indexing level*. Terdapat dua masukan yang akan diolah yaitu *100000.siftgeo* yang berisikan fitur-fitur dari *dataset Holiday* dan *flickr60k.siftgeo* yang berisikan fitur-fitur dari *dataset Flickr60k*. *100000.siftgeo* berfungsi sebagai *database* karena ukurannya lebih besar dibandingkan dengan *flickr60k.siftgeo*, sedangkan *flickr60k.siftgeo* berfungsi sebagai data uji atau bisa disebut dengan *distractor* karena data yang terdapat dalam *flickr60k.siftgeo* belum tentu seluruhnya terdapat dalam *100000.siftgeo*. Gambar 3.1 menjelaskan mengenai perancangan sistem secara umum.



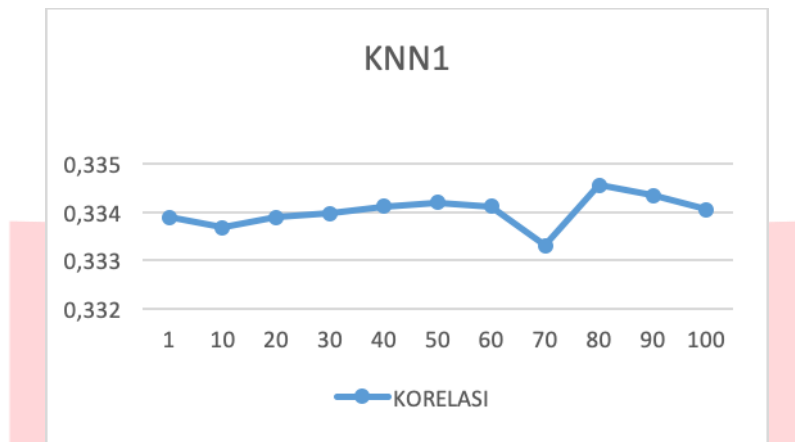
Gambar 3.1 Diagram Blok Sistem Secara Umum

### B. Parameter Performansi Sistem

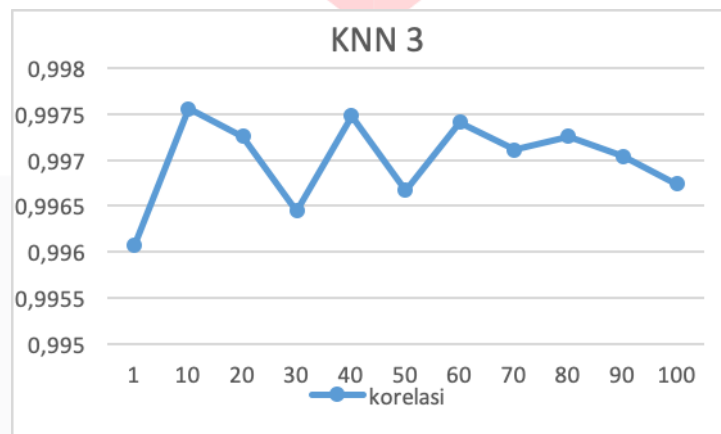
Setelah dilakukan tahap proses dari sistem, maka tahap selanjutnya yaitu akan dilakukan evaluasi terhadap performansi sistem. Evaluasi ini bertujuan untuk mengetahui kinerja dari sistem yang telah dirancang, apakah layak atau tidak. Performansi sistem dapat diukur dengan menggunakan parameter nilai *k* pada *nearest neighbor*, dan mengganti *codebook* yang telah didapat dalam tahap sebelumnya. Nilai *k* dan *codebook* akan dilihat pengaruhnya pada hasil *Korelasi*.

#### 4. Hasil Pengujian

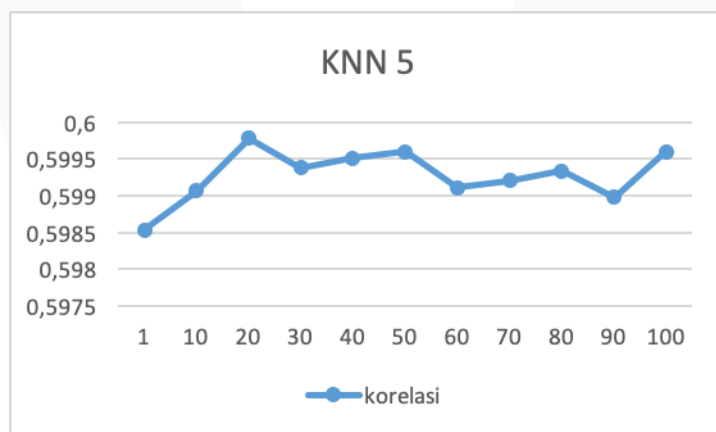
##### A. Pengaruh Nilai *Nearest Neighbor* dan *Codebook*



Gambar 4. 1 Grafik nilai  $k=1$



Gambar 4. 2 Grafik nilai  $k=2$



Gambar 4. 3 Grafik nilai  $k=3$

Berdasarkan dengan grafik yang ditampilkan, dapat disimpulkan bahwa nilai  $k$  yang dapat menghasilkan nilai korelasi yang paling baik adalah saat  $k = 3$ . Sedangkan nilai  $k$  yang mempunyai nilai korelasi yang paling rendah adalah saat  $k = 1$ . Hal ini disebabkan jika diambil nilai  $k=1$  maka *nearest neighbor* menjadi terlalu sedikit, sedangkan jika nilai  $k=5$  maka *nearest neighbor* terlalu banyak, sehingga nilai  $k=3$  merupakan jumlah *nearest neighbor* yang dapat menghasilkan nilai korelasi maksimum. Nilai korelasi tertinggi yaitu senilai 0.997556 saat  $k=3$  dan *codebook* 10. Hal ini terlihat dari ciri yang diamati, ciri pada *100000.siftgeo* banyak ditemukan pada ciri

yang terdapat dalam *database Flickr60k* yang terkandung dalam *codebook* 10, sedangkan ciri pada *100000.siftgeo* ditemukan lebih sedikit pada *database Flickr60k* yang terkandung dalam *codebook* 70 sehingga menghasilkan nilai korelasi yang rendah.

## 5. Kesimpulan

Pada Tugas Akhir ini menggunakan metode *c-MI* untuk *image retrieval*, dimana setiap *keypoint* yang terdapat pada citra dideskripsikan oleh SIFT dan *color feature*, metode ini bekerja pada *index level*. Berdasarkan penelitian yang telah dilakukan pada Tugas Akhir ini, sistem dapat dijalankan dengan harapan menghasilkan nilai korelasi = 1. Nilai korelasi terendah yang didapat adalah sebesar 0.333309 dengan parameter  $k=1$  dan menggunakan *codebook* 70. Nilai korelasi tertinggi yang didapat adalah sebesar 0.997556 dengan parameter  $k=3$  dan menggunakan *codebook* 10. Sehingga menandakan bahwa ciri pada *dataset holiday* menyerupai ciri pada *database* yang terkandung dalam *codebook* 10.

## Daftar Pustaka

- [1] I. Putra, "Analisis dan Implementasi Scale Invariant Feature Transform (SIFT) pada Content Based Image Retrieval," pp. 1-4, 2011.
- [2] D. G. Lowe, "Object Recognition from Local Scale Invariant Features," Proceedings of the International Conference on Computer Vision, pp. 1150-1157, 1999.
- [3] L. e. a. Zheng, "Packing and Padding: Coupled Multi-index for Accurate Image Retrieval," Conference on Computer Vision and Pattern Recognition, pp. 1947-1954, 2014.
- [4] J. e. a. Tang, "Neighborhood Discriminant Hashing for Large Scale Image Retrieval," Transaction on Image Processing, pp. 1-12, 2015.
- [5] T. Sutoyo, E. Mulyanto, V. Suhartono and O. N. a. Wijanarto, Teori Pengenalan Citra Digital, Semarang: Andi, 2009.
- [6] F. L. Gaol, *Recent progress in data engineering and internet technology*. Springer, 2012.