BAB 2
TINJAUAN PUSTAKA

2.1 Kartu Tanda Penduduk Elektronik

Kartu Tanda Penduduk Elektronik atau E-KTP adalah Kartu Tanda Penduduk yang dibuat secara elektronik, sehingga dalam penggunaannya berfungsi secara komputerisasi, dan merupakan dokumen kependudukan yang memuat system keamanan/pengendalian baik dari sisi administrasi ataupun teknologi informasi dengan berbasis pada *database* kependudukan nasional.

E-KTP berisikan identitas data diri dari setiap orang, maka dalam E-KTP terdapat sebuah nomor unik yang disebut Nomor Induk Kependudukan (NIK).

E-KTP digunakan dalam setiap pelayanan masyarakat yang membutuhkan identitas dan data diri setiap orang. Nomor NIK yang terdapat pada E-KTP dijadikan dasar dalam penerbitan Paspor, Surat Izin Mengemudi (SIM), Nomor Pokok Wajib Pajak (NPWP), Polis Asuransi, Sertifikat atas Hak Tanah dan penerbitan dokumen identitas lainnya (Pasal 13 UU No. 23 Tahun 2006 tentang Adminduk).[1]

NIK mempunyai karakter yang unik untuk membedakan satu sama lain. NIK tersusun atas 16 digit dengan format :

AABBCCDDEEFFGGGG

Adapun penjelasan mengenai format di atas adalah sebagai berikut:

AA (1-2)	: kode provinsi NIK diterbitkan
BB (3-4)	: kode kabupaten/kota NIK diterbitkan
CC (5-6)	: kode kecamatan NIK diterbitkan
DD (7-8)	: tanggal lahir, (jika wanita, tanggal lahir ditambah 40)
EE (9-10)	: bulan lahir
FF (11-12)	: dua angka terakhir tahun lahir
GGGG (13-16)	: nomor komputerisasi, nomor yang diatur secara random oleh komputer agar berbeda satu sama lain. Namun untuk kepala keluarga biasanya ditulis xx01, untuk anak pertama xx02 dan seterusnya.

2.2 Ekstraksi Informasi

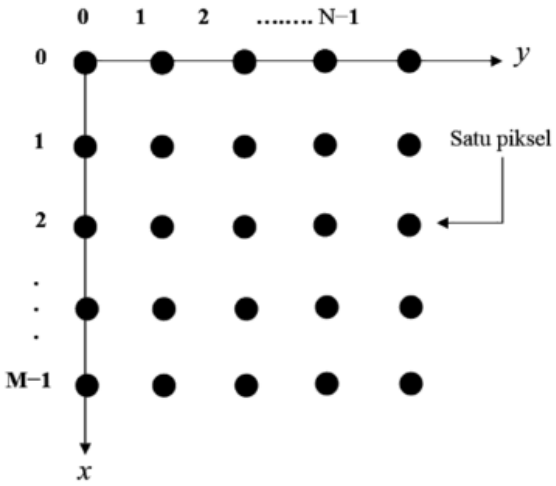
Ekstraksi informasi merupakan proses mengubah informasi yang tidak terstruktur yang terdapat pada sebuah citra atau teks dan dijadikan ke dalam data terstruktur [2]. Data terstruktur adalah data yang berkaitan dengan fakta di dunia nyata yang terdapat dalam sebuah citra atau dokumen seperti entitas dan atributnya berupa kejadian apa yang sedang terjadi, siapa aktor dalam kejadian, kapan dan di mana kejadian tersebut, atau kenapa dan bagaimana kejadian tersebut bisa terjadi [19]. Ekstraksi informasi ini merupakan bagian dari *Natural Language Processing* (NLP) [2]. Informasi dapat diatur sesuai dengan target yang akan dicapai, sehingga informasi yang berkaitan sesuai dengan apa yang dibutuhkan.

Ekstraksi informasi merupakan jenis pencarian yang menentukan informasi spesifik dari sebuah citra atau dokumen yang akan di ekstrak. Kegiatan tersebut sangat fleksibel dengan metode apapun yang bertujuan untuk mengekstrak informasi.

Ekstraksi informasi mempunyai dua pendekatan yang paling mendasar yaitu, pengetahuan teknik dan pelatihan otomatis. Pengetahuan teknik sangat berperan kepada ahli guna mengembangkan aturan untuk memecahkan masalah. Pelatihan otomatis merupakan salah satu pendekatan dari ekstraksi informasi. Pada kegiatan pendekatan tersebut tidak memerlukan seorang ahli untuk melakukan aturan penulisan ekstraksi, tetapi membutuhkan seseorang yang mengetahui pada bidang tertentu untuk mendapatkan informasi yang di ekstrak [20].

2.3 Citra Digital

Citra merupakan suatu gambaran dari suatu objek. Citra digital merupakan citra yang dapat diolah kembali oleh perangkat pengolahan citra seperti komputer [21]. Citra digital didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom (*pixel*), dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari sebuah citra pada titik tersebut [22]. Gambar 2.1 menunjukkan posisi koordinat dari citra digital.



Gambar 2.1 Sistem Koordinat Citra Digital

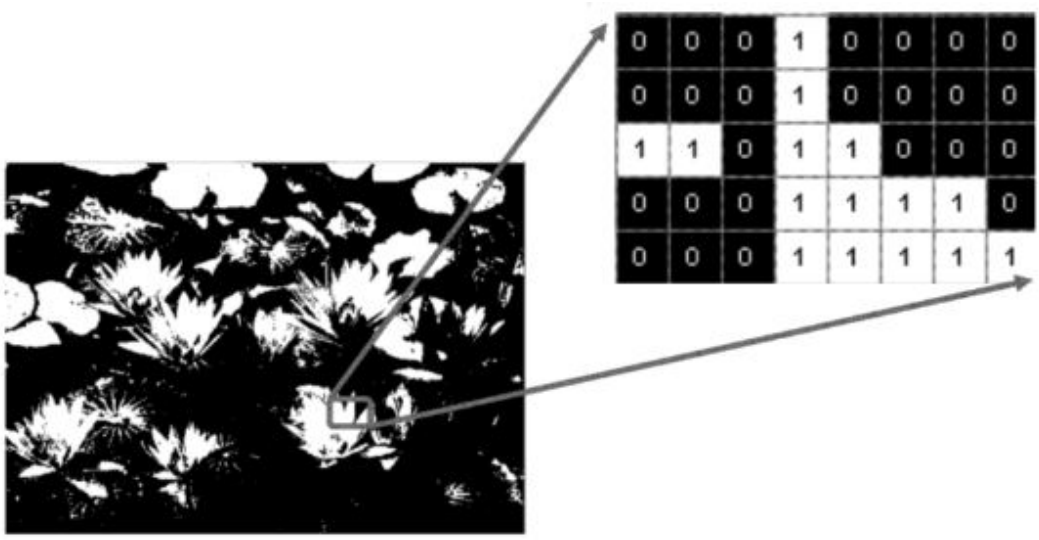
Citra digital dapat ditulis dalam bentuk matriks sebagai berikut:

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

Citra digital pada umunya dibagi menjadi tiga jenis yaitu citra biner, citra *grayscale*, dan citra warna. Masing-masing dari citra tersebut mempunyai definisi yang berbeda yaitu:

2.3.1 Citra Biner

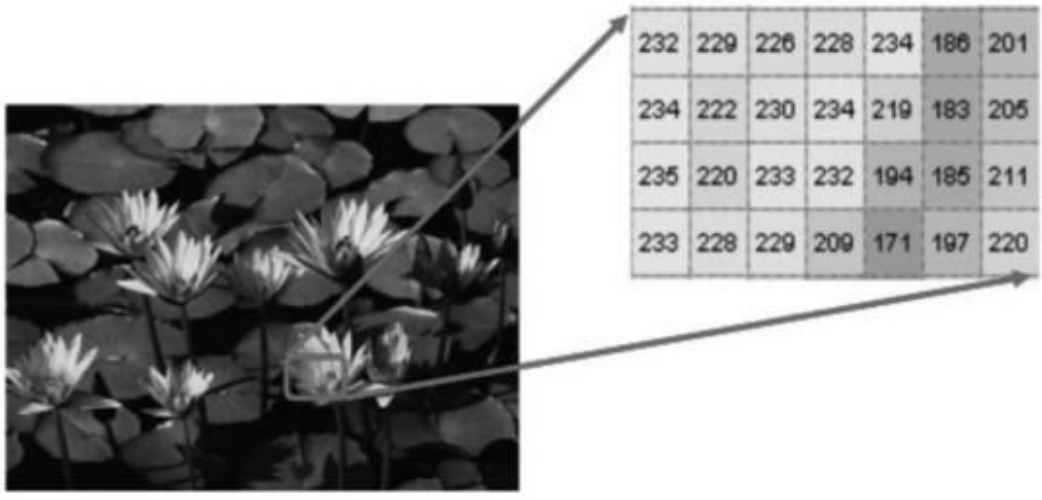
Citra biner disebut sebagai citra B&W (*Black and White*) atau monokrom, karena dalam citra biner hanya membutuhkan 1bit untuk mewakili nilai setiap *pixel*. Sehingga, setiap *pixel* hanya mempunyai dua kemungkinan yaitu hitam (0) dan putih (1) [21], [22].



Gambar 2.2 Citra Biner dengan Nilai *Pixel* 0 atau 1

2.3.2 Citra *Grayscale*

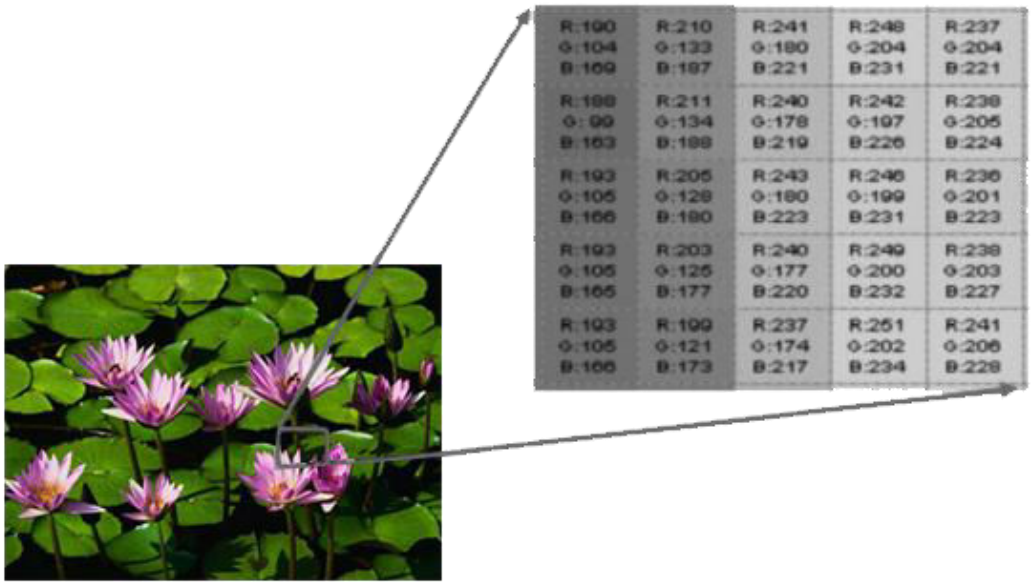
Citra *grayscale* adalah sebuah citra yang memiliki warna dari hitam, keabuan, dan putih. Citra *grayscale* merupakan citra digital yang mempunyai satu nilai kanal pada setiap *pixel*nya (*RED = GREEN = BLUE*) untuk menunjukkan tingkat intensitas. Citra *grayscale* mempunyai nilai intensitas setiap *pixel* yang berkisar antara 0 sampai dengan 255 (256 kombinasi warna keabuan) dan memunyai kedalaman warna 8bit [21], [22].



Gambar 2.3 Citra *Grayscale* dengan Nilai *Pixel* antara 0 sampai dengan 255

2.3.3 Citra Warna

Citra warna merupakan citra yang masing-masing *pixel* mempunyai tiga komponen warna yang spesifik, yaitu komponen merah (*red*), hijau (*green*), biru (*blue*). Warna setiap *pixel* ditentukan dari kombinasi ketiga komponen warna. Format *file* grafis memiliki citra warna 24bit, yang berasal dari setiap komponen warna masing-masing 8 bit (0 sampai dengan 255). Hal tersebut menyebabkan citra warna mempunyai 16.777.216 variasi warna [21], [22].



Gambar 2.4 Citra Warna

2.4 Pengolahan Citra Digital

Pengolahan citra digital merupakan disiplin ilmu yang mempelajari tentang teknik-teknik untuk mengolah data citra digital untuk memperbaiki kualitas citra. Pengolahan citra digital dilakukan dengan menggunakan pengolahan berdasarkan warna. Model RGB, CMY, HSI, HSV, dan *normalized* RGB merupakan beberapa analisis warna dari pangolahan data citra digital [23]. Menurut Putra (2010), pengolahan citra dikategorikan kedalam tiga kategori tingkatan yaitu rendah, menengah, dan tinggi [22].

1. Pengolahan tingkat rendah, pengolahan yang melibatkan operasi-operasi sederhana seperti pengolahan citra untuk mengurangi derau, pengaturan

kontras, dan pengaturan ketajaman citra. Pengolahan pada tingkat ini memiliki *input* dan *output*.

2. Pengolahan tingkat menengah, yaitu pengolahan yang melibatkan operasi-operasi seperti segmentasi dan klasifikasi citra. Pengolahan pada tingkat ini melibatkan *input* berupa citra dan *output* berupa fitur citra yang dipisahkan dari citra *input*.
3. Pengolahan tingkat tinggi, yaitu pengolahan yang melibatkan proses pengenalan dan deskripsi citra.

2.5 *Optical Character Recognition*

Optical Character Recognition (OCR) merupakan cabang dari *artificial intelligence* dan *computer vision*. OCR merupakan sebuah aplikasi komputer yang digunakan untuk membaca citra huruf ataupun angka yang dikonversi kedalam bentuk tulisan. OCR ini dapat meningkatkan kemampuan dan kecerdasan komputer, karena sistem OCR ini sangat membantu usaha digitalisasi informasi dan pengetahuan [3], [24].

Hasil dari OCR merupakan teks sesuai dengan gambar masukkan di mana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan pada tahap *preprocessing* [25]. Secara umum tahapan OCR dapat dilihat pada Gambar 2.5, berikut penjelasannya.

Gambar 2.5 Proses OCR Secara Umum

Dari Gambar 2.5, citra digital sebelum dikenali menjadi teks, karakter sebuah citra dikenali satu persatu.

a. *File Input*

   File Input merupakan data citra yang dimasukkan ke dalam sistem dengan format *.jpg, *.bmp, atau *.png.

b. *Preprocessing*

   Preprocessing merupakan proses pengolahan data citra untuk menghilangkan bagian yang tidak diperlukan.

c. Segmentasi

   Segmentasi merupakan proses pemisahan area yang diamati pada tiap karakter yang dideteksi.

d. Normalisasi

   Normalisasi merupakan proses untuk merubah dimensi yang diamati tiap karakter dan ketebalan karakter.

e. Ekstraksi Ciri

   Ekstraksi ciri yaitu proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati

f. *Recognition*

   Recognition yaitu proses untuk mengenali karakter yang diamati.

2.6 *Preprocessing*

Preprocessing merupakan salah satu tahapan yang digunakan untuk menghilangkan *noise* pada data citra yang akan digunakan. Pada tahap *preprocessing* ini terdiri dari *resize, grayscale,* dan *thresholding*. Proses ini menghasilkan data citra dengan warna biner yaitu hitam (0) dan putih (1) [25].

2.6.1 *Grayscale*

Grayscale merupakan hasil konversi dari citra yang memiliki 3 komponen warna (RGB), dari komponen tersebut kemudian dikonversikan menjadi citra *grayscale* [26]. Proses ini merubah RGB kanal menjadi citra dengan 1 kanal, sehingga proses ini bertujuan untuk merubah citra berwarna menjadi citra yang

berskala keabuan. *Grayscale* pada umumnya menggunakan jumlah bit sebesar 8 bit [27], [28]. Citra *grayscale* dapat didefinisikan sebagai berikut:

$$f(x,y) = (0.2989 \times R) + (0.5870 \times G) + (0.1141 \times B)$$

2.6.2 Binerisasi

Binerisasi adalah proses yang bertujuan untuk merubah data citra dari citra *grayscale* menjadi citra biner. Citra biner mempunyai dua nilai yaitu hitam (0) dan putih (1). Untuk melakukan proses binerisasi diperlukan nilai ambang dari *threshold*. Tujuan utamanya yaitu untuk memisahkan objek dengan *background*.

Threshold pada citra *g(x,y)* di mana *f(x,y)* mewakili citra, kemudian T sebagai nilai dari *threshold*. Citra *threshold* dapat didefinisikan sebagai berikut [27]:

$$g(x,y) = \begin{cases} 1 \ if & f(x,y) \geq T \\ 0 \ if & f(x,y) < T \end{cases}$$

Metode yang digunakan dalam proses binerisasi adalah metode *threshold otsu.* Metode *otsu* merupakan metode yang dapat secara otomatis menemukan nilai ambang batas pada citra digital dengan melakukan analisis diskriminan terhadap tingkat keabuan histogram yang diharapkan dapat memaksimalkan pemisahan objek dengan latar belakangnya. Nilai ambang yang dicari dari suatu citra *grayscale* dinyatakan dengan *k* yang bernilai antara 1 sampai dengan *L*, dengan nilai *L* = 255. Tahapan dari metode *otsu* adalah sebagai berikut [29]:

1. Menentukan probabilitas setiap pixel pada level ke *i* dapat dinyatakan sebagai berikut :

   $$p_i = \frac{n_i}{N}$$

   Keterangan :

   $p_i = probabilitas\ pixel$

   $n_i = jumlah\ pixel\ pada\ level\ ke\ i$

   $N = total\ jumlah\ pixel\ pada\ citra$

2. Menentukan nilai *Zero cumulative moment, First cumulative moment*, dan total *mean* berturut-turut dapat dinyatakan dengan persamaan berikut :
   a. *Zero cumulative moment*

$$\omega(k) = \sum_{i=1}^{k} p_i$$

   b. *First cumulative moment*

$$\mu(k) = \sum_{i=1}^{k} i.p_i$$

   c. *mean*

$$\mu_T = \sum_{i=1}^{L} i.p_i$$

3. Menentukan nilai ambang *k* yang dapat ditentukan dengan menggunakan persamaan berikut :

$$\sigma_B^2(k^*) = \max_{1 \le k \le L} \sigma_B^2(k) \text{ dengan } \sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1-\omega(k)]}$$

2.6.3 *Resize*

Resize merupakan proses mengubah ukuran dari data citra menjadi lebih kecil maupun lebih besar. Pada proses memperkecil ukuran citra, dilakukan dengan cara mengurangi jumlah *pixel* yang terdapat pada citra [9]. *Resize* dilakukan untuk mengurangi kinerja dari komputer sehingga proses berjalan lebih cepat dari semula, kemudian *resize* juga memakan lebih sedikit ruang memori [26].

2.6.4 Segmentasi Kontur

Segmentasi merupakan proses pembagian sebuah citra menjadi daerah pilihan dari citra secara keseluruhan. Hal-hal yang mempengaruhi proses segmentasi agar bisa dilakukan yaitu tekstur, kecerahan, serta intensitas jumlah *pixel* [30].

Rangkaian pixel-pixel yang membentuk daerah (*regian boundary*) disebut dengan kontur (*countur*). Metode dalam komputasi geometris algoritma, untuk

ekstraksi kontur dilakukan dalam dua tahap, yang pertama adalah konversi *input* dan algoritma geometrik dan tahap kedua menemukan titik-titik dari kontur yang saling terhubung menggunakan algoritma geometrik [31].

Tahap kedua memiliki tiga langkah, yang pertama untuk pengelompokan titik digunakan teknik pengelompokan, setelah itu setiap titik dihubungkan berdasarkan kedekatan dan orientasinya, serta untuk menyederhanakan jalur dilakukan dengan mengurangi jumlah poin yang terdapat di dalam tepi [31].

Hasil dari kontur dapat berupa senarai tepi (edge list) atau berupa kurva. Gambaran hasil dari kontur ke dalam kurva merupakan hasil yang kompak dan mangkus untuk analisis citra. Misalnya, rangkaian pixel tepi yang membentuk garis dapat digambarkan hanya dengan sebuah persamaan garis lurus. Hasil yang demikian menyederhanakan perhitungan selanjutnya seperti arah dan panjang garis [31].

2.7 Ekstraksi Ciri *Zoning*

Ekstraksi ciri dilakukan untuk mengenali ciri dari setiap karakter. Ekstraksi ciri *Zoning* adalah salah satu metode ekstraksi ciri di mana garis besar dari metode ini adalah membagi citra pada sejumlah zona yang sama untuk dikenali ciri dari setiap karakternya yang selanjutnya menghasilkan sebuah nilai untuk diproses pada tahapan klasifikasi [32], [33].

Berikut merupakan tahapan-tahapan dalam algoritma *Zone Centroid Zone* (ZCZ) [32], [33]:

1. Bagi citra masukkan ke dalam sejumlah *n* bagian yang sama
2. Hitung *centroid* dari masing-masing zona

$$x_c = \frac{(x_1.p_1 + x_2.p_2 + \cdots + x_n.p_n)}{(p_1 + p_2 + \cdots + p_n)}$$

$$y_c = \frac{(y_1.p_1 + y_2.p_2 + \cdots + y_n.p_n)}{(p_1 + p_2 + \cdots + p_n)}$$

Keterangan :

x_c = Centroid koordinat *x*

y_c = Centroid koordinat *y*

x_n = Koordinat *x* dari pixel - *n*

y_n = Koordinat *y* dari pixel - *n*

p_c = Nilai pixel ke – *n*

3. Hitung jarak antara *centroid* masing-masing zona dan *pixel* yang ada di zona

$$d(P, C) = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

Keterangan :

d = jarak antara dua titik

P = Koordinat titik hitam

C = Koordinat pixel

x_p = Koordinat *x* titik berat

y_p = Koordinat *y* titik berat

x_c = Koordinat *x* titik pixel

y_c = Koordinat *y* titik pixel

4. Ulangi langkah ke 3 untuk seluruh *pixel* yang ada di zona
5. Hitung rata-rata jarak antara titik-titik tersebut

$$Rataan\ Jarak = \frac{(jarak(x_1.y_1) + jarak(x_2.y_2) + \cdots + jarak(x_n.y_n))}{banyak\ titik\ pada\ zona}$$

6. Ulangi langkah 3-5 untuk setiap zona secara berurutan
7. Hasilnya adalah *n* fitur yang akan digunakan dalam klasifikasi dan pengenalan

2.8 *Reduced Support Vector Machine*

Algoritma *Reduced Support Vector Machine* (RSVM) merupakan algoritma yang bertujuan untuk mengatasi masalah komputasi dalam jumlah data yang besar. Algoritma RSVM ini menggunakan 10% data secara acak dari jumlah data yang digunakan. RSVM merupakan algoritma yang menghasilkan permukaan pemisah berbasis kernel nonlinier yang membutuhkan sedikitnya 1% dari keseluruhan *dataset* untuk evaluasi eksplisitnya. Seluruh *dataset* digunakan sebagai pembatas dalam masalah optimisasi dengan sangat sedikit variabel yang sesuai dengan 1% dari data yang disimpan. Sisa data dapat dibuang setelah menyelesaikan masalah optimisasi. Langkah-langkah dalam algoritma RSVM dapat dituliskan sebagai berikut [14]:

1. Pilih matriks subset acak $\bar{A} \in R^{\bar{m}xn}$ dari data awal $A \in R^{mxn}$ secara acak. Biasanya $\bar{m}$ adalah 1% hingga 10% dari m.
2. Menyelesaikan persamaan *Smooth Support Vector Machine* (SSVM) yang telah dilakukan modifikasi, dan dapat diselesaikan dengan menggunakan algoritma *Newton-Armijo* [34] di mana A' saja yang digantikan oleh $\bar{A}'$ dengan $\bar{D} \subset D$:

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \frac{v}{2} ||p(e - D(K(A,\bar{A}')\bar{D}\bar{u} - e\gamma),\alpha)||_2^2 + \frac{1}{2}(\bar{u}'\bar{u} + \gamma^2)$$

3. Bidang pemisah dengan A' diganti dengan $\bar{A}'$ sebagai berikut:

$$K(x', \bar{A}')\bar{D}\bar{u} = \gamma$$

di mana $(\bar{u}, \gamma) \in R^{m+1}$ adalah solusi unik dari persamaan, dan $x \in R^n$ adalah variabel ruang *input* bebas dari titik baru.

4. Titik *input* baru $x \in R^n$ diklasifikasikan ke dalam kelas +1 atau −1 tergantung pada fungsi:

$$f(x) = (K(x', \bar{A}')\bar{D}\bar{u} - \gamma)$$

dengan *f(x)* masing-masing nilainya adalah +1 atau 0.

2.9 *K-Fold Cross Validation*

Cross-Validation atau disebut juga dengan estimasi rotasi adalah sebuah teknik validasi model untuk menilai hasil statistik analisis akan menggeneralisasi kumpulan data. Teknik utamanya digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model ketika dijalankan. Salah satu teknik dari validasi silang adalah *k-fold cross validation*, di mana dapat melakukan pemecahan data menjadi *k* bagian set data dengan ukuran yang sama. Pelatihan dan pengujian dilakukan sebanyak *k* kali [29].

2.10 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman tingkat inggi (*high level language*) yang dikembangkan oleh Guido van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinya pada tahun 1991. Python dirancang agar memberikan kemudahan kepada programmer baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program khususnya dalam hal kompatibilitas dengan sistem [35]. Bahasa pemrograman python dirancang agar mudah dibaca oleh orang. Sebagai implementasinya, penggunaan spasi sangat berpengaruh terhadap program, sehingga penggunaan spasi tidak dapat dilakukan dengan sembarangan. Python dapat digunakan pada berbagai platform seperti Windows, Mac OS, dan Linux [36].

Python memiliki beberapa kelebihan sehingga bahasa pemrograman python banyak digunakan oleh kalangan *engineer*. Berikut merupakan kelebihan yang dimiliki bahasa pemrogram python: [35], [37]

a. Mudah dipelajari karena perintah-perintah yang terdapat dalam python dirasa cukup singkat dan mudah dimengerti.
b. *Freeware* dan *Open source*, sehingga kita tidak perlu khawatir dengan masalah lisensi.
c. Pemrograman cukup *powerfull*, dapat membuat berbagai aplikasi dekstop, network hingga website. Bahkan python terkenal dengan pembuatan aplikasi *hacking*.

d. Portable dan dapat digunakan diberbagai sistem operasi, baik Windows, Linux, ataupun Mac OS.
e. Python kaya akan dukungan *library.*
f. Penulisan kode lebih efisien dibandingkan dengan bahasa pemrograman lain.
g. Mempunyai dukungan terhadap ekosistem *Internet of Things* (*IoT*) dengan sangat baik.
h. Pemrograman berorientasi objek (*Object Oriented Programming*).