BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian di bidang pengenalan huruf, karakter atau tulisan tangan sangat luas objek dan pemanfaatannya, misalnya untuk mengenal tulisan tangan ekspresi matematika (Genoe *et all,* 2006; MacLean dan Labahn, 2010), pengenalan tulisan tangan untuk pembuktian *authorship* identifikasi penulis (Muda *et all*, 2009), pengenalan tulisan tangan dari layar sentuh sebuah *gadget* (Aminulloh dkk, 2008). Penelitian pengenalan tulisan tangan juga telah dilakukan pada variasi tulisan di beberapa belahan dunia, misalnya pengenalan tulisan tangan Tamil (Kannan dan Prabhakar, 2008; Shanthi dan Duraiswamy, 2007; Venkatesh dan Sureshkumar, 2009), pengenalan tulisan Devnagari (Atul dan Mishra, 2007; Shrivastava dan Gharde, 2010), pengenalan tulisan Arab (Al-Alaoui *et all*, 2009; Al-Jawfi *et all*, 2009; Ali Abed *et all*, 2010; Abed *et all,* 2010; Abu-Ain *et all*, 2011; Affar *et all*, 2009; Amin dan Darwish, 2006; Khemakhem dan Belghith, 2009; Leila dan Mohammed, 2007; Njah *et all*, 2007), pengenalan tulisan China (Su *et all*, 2008), pengenalan pola huruf Jepang (Kana) (Wirayuda dkk, 2008), pengenalan tulisan tangan Azerbaijani (Aliyeva dan Ismayilov, 2008), pengenalan tulisan tangan Persia (Baghshah *et all*, 2005), pengenlanan tulisan tangan Bengali (Sarowar *et all,* 2009), pengenalan tulisan tangan Sinhala (Kodituwakku dan Nilanthi, 2010), pengenalan tulisan tangan Bangla (Naser *et all,* 2009), pengenalan karakter Urdu Nastalique (Sattar *et all,* 2009). Di Indonesia juga terdapat variasi tulisan daerah.

Beberapa diantaranya sudah diangkat menjadi objek penelitian pengenalan tulisan tangan, yaitu pengenalan huruf Bali (Wirayuda dkk, 2009), pengenalan tulisan Jawa-Hanacaraka (Winardi dkk, 2010) yang melakukan pengembangan metode pada tahap *pre-processing*, pengenalan tulisan tangan aksara Sunda (Mubarok dkk, 2010), pengenalan huruf Lontara Bugis-Makasar (Alwi, 2009) dan pengenalan alfabet Batak Toba (Panggabean dan Rønningen, 2009) menggunakan metode *simplified chain code*.

Selain objek tulisan yang dikenali beragam, penelitian pengenalan tulisan tangan juga dilakukan dengan menggunakan metode yang beragam, baik pada tahap akuisisi data, proses awal, ekstraksi ciri karakter dan bahkan pada tahap klasifikasi. Munggaran dkk (2009) melakukan pengenalan tulisan secara *online* (data diperoleh secara langsung dari alat input berupa *keyboard*, *mouse* atau *touch pen*) untuk akuisisi karakter dengan tepat dengan segmentasi berdasarkan karakteristik perubahan koordinat Y dari setiap pembentukan huruf. Gatos *et all* (2006) melakukan pengenalan huruf kursif dengan akuisisi data secara *offline* (data diperoleh dari hasil *scan* citra atau kamera digital) dengan kombinasi normalisasi dan ekstraksi ciri *hybrid*. Wu dan Yu (2008) memaparkan tentang pemanfaatan alat *touchless* (akuisisi data menggunakan kamera *web*) untuk pengenalan tulisan tangan menggunakan JST dan diperoleh hasil pengenalan dan kecepatan yang cukup baik. Nuryuliani dkk (2009 dan 2010) yang melakukan pengenalan huruf Latin dengan menggunakan pola segmen untuk ekstraksi cirinya. Razak *et all* (2008) melakukan kajian beberapa metode segmentasi pada pengenalan tulisan. Senouci *et all* (2007) memaparkan metode segmentasi untuk

pengenalan kata yang menghasilkan 81.05% akurasi dan secara keseluruhan akurasi pengenalan yang dicapai adalah 91%.

Pengenalan tulisan tangan merupakan salah satu contoh aplikasi pengenalan pola yang cukup kompleks. Tidak adanya fungsi matematika yang jelas untuk menghasilkan translasi yang diinginkan membuat penerjemahan karakter atau tulisan tangan menjadi sesuatu yang cukup sulit, karena untuk menghasilkan translasi salah satu caranya adalah dengan korelasi piksel demi piksel yang membutuhkan waktu yang relatif lama. Selain itu, tidak memadainya basis pengetahuan yang diakibatkan aturan-aturan pengetahuan yang sulit dirumuskan dan keterlibatan data yang mengandung *noise* dan berjumlah cukup besar (perlu pemrosesan cepat), membuat penelitian di bidang pengenalan karakter atau tulisan tangan membutuhkan metode jaringan saraf tiruan (JST) untuk penyelesaian untuk klasifikasinya (Puspitaningrum, 2009). JST memperoleh pengetahuan melalui proses pembelajaran. Venkatesh dan Sureshkumar (2009) menggunakan metode JST *Kohonen's Self Organizing Map* untuk pengenalan tulisan tangan Tamil. Mubarok dkk (2010) menggunakan metode *Kohenen Neural Network* untuk pengenalan tulisan tangan aksara Sunda. Asworo (2009) melakukan penelitian pengenalan tulisan angka dan abjad Latin dengan membandingkan kinerja metode JST *Kohonen* dan *Learning Vector Quantization (LVQ)* yang membuktikan akurasi hasil pengenalan pada *LVQ* lebih tinggi (86%) dari metode *Kohonen* (68%). Wirayuda dkk menggunakan *Learning Vector Quantization (LVQ)* untuk pengenalan pola huruf Jepang (Kana) (2008) dan pengenalan huruf Bali (2009) dengan menghasilkan persentase pengujian

70% pada data uji dengan penulis yang berbeda dan di atas 80% dengan penulis yang tulisannya pernah menjadi data *training*. Ismail *et all* (2010) mengangkat penelitian identifikasi tanda tangan menggunakan metode ekstraksi ciri *fourier descriptor* dan *chain codes* dan metode untuk klasifikasinya menggunakan JST *multilayer feed forward*. Al-Jawfi *et all* (2009) melakukan pengenalan tulisan tangan huruf Arab menggunakan JST LeNet yang terdiri dari dua tahap, yaitu tahap untuk mengenal bentuk utama dari huruf dan tahap untuk pengenalan titik.

Salah satu metode JST yang sering digunakan untuk pengenalan tulisan tangan adalah *Backpropagation*. Metode *Backpropagation* sangat baik dalam menangani masalah pengenalan pola-pola kompleks, sehingga sering pula dipilih untuk kasus pengenalan karakter (Puspitaningrum, 2006). Aplikasi memakai jaringan ini untuk masalah yang melibatkan pemetaan sekumpulan masukan terhadap sekumpulan target keluaran, jadi *Backpropagation* termasuk kategori jaringan dengan pelatihan terbimbing (Setiawan, 2005; Puspitaningrum, 2006). Park *et all* (2008) dan Salameh dan Otair (2008) menggunakan *Backpropagation* untuk membangun aplikasi OCR (*Optical Character Recognition*). Sarowar *et all* (2009) mengembangkan pengenalan tulisan tangan Bengali menggunakan heuristik untuk membuat pelatihan JST *Backpropagation* semakin baik. Mathur *et all* (2008) mengkombinasikan JST *Backpropagation* dan algoritma genetika yang menghasilkan efisiensi 71%. JST *Backpropagation* memiliki waktu pelatihan yang lambat, karena gradien *error* unit-unit tersembunyi diturunkan dari penyiaran kembali *error-error* yang diasosiasikan dengan unit-unit output (perambatan balik). Hal tersebut bertujuan untuk mendapatkan kesetimbangan

antara respon yang benar terhadap pola pelatihan (kemampuan memorisasi) dan respon yang baik terhadap pola-pola input baru (kemampuan generalisasi) (Puspitaningrum, 2006). Akan tetapi terdapat salah satu teknik untuk mempercepat pelatihan, yaitu dengan penambahan momentum seperti pada penelitian Otair dan Samlameh (2008) mengembangkan pengenalan tulisan tangan yang menghasilkan proses pengenalan yang lebih cepat dibandingkan *Backpropagation* tanpa momentum. Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Disamping arsitektur jaringan dan algoritma pelatihan/pembelajaran, sebuah JST ditentukan oleh fungsi aktifasi / fungsi transfer yang digunakan. Pada JST terbimbing, termasuk metode *Backpropagation* pada umumnya menggunakan fungsi aktifasi sigmoid untuk membangkitkan nilai output. Fungsi sigmoid biner memiliki beberapa karakteristik penting untuk jaringan *Backpropagation*, yaitu kontinu, dapat diturunkan dan tidak menurun secara monoton, dan turunannya mudah untuk dilakukan perhitungan (Puspitaningrum, 2006, Siang, 2009). Brüderle *et all* (2006) melakukan pengenalan angka dengan menggunakan fungsi aktifasi sederhana dimana akurasi pengenalan dapat diatur berdasarkan parameter fungsi.

Dalam penelitian ini dilakukan pengenalan tulisan tangan aksara Batak Toba dimana akuisisi citra dilakukan dengan menggunakan kamera digital untuk meng-*capture* tulisan tangan aksara Batak Toba yang ditulis di kertas. Citra tersebut kemudian diolah, yaitu dengan mengubahnya mejadi citra biner dan

ditransformasi menggunakan *Wavelet* sehingga menjadi masukan untuk proses klasifikasi menggunakan metode *Bakpropagation*.

2.2. Hipotesis

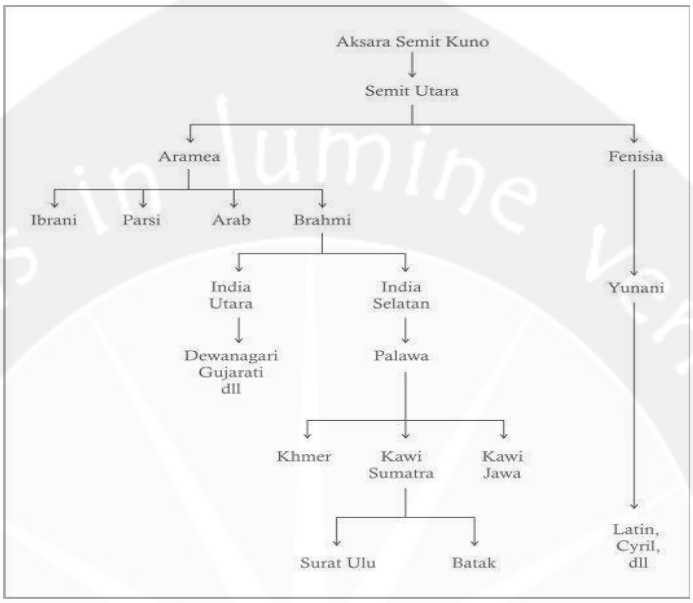
Berdasarkan beberapa literatur yang telah dikaji di atas dan fakta-fakta yang diperoleh pada penelitian-penelitian sebelumnya, maka dalam penelitian ini dapat diambil hipotesis bahwa metode JST *Backpropagataion* merupakan metode yang tepat atau cocok untuk menyelesaikan permasalahan pada penelitian ini. Permasalahan dalam penelitian ini adalah pengenalan tulisan tangan aksara Batak Toba, dimana tidak tersedia fungsi matematis yang jelas untuk menghasilkan translasi yang diinginkan, tidak memadainya basis pengetahuan yang diakibatkan aturan-aturan pengetahuan yang sulit dirumuskan dan keterlibatan data yang mengandung *noise* dan berjumlah cukup besar (perlu pemrosesan cepat) dimana permasalahan seperti ini cocok diselesaikan dengan metode JST *Backpropagation*.

2.3. Landasan Teori

2.3.1. Aksara Batak Toba

Aksara (*surat*) Batak termasuk keluarga tulisan India. Aksara India yang tertua adalah aksara Brahmi yang menurunkan dua kelompok tulisan, yakni kelompok India Utara (Aksara Nagari) dan kelompok India Selatan (Aksara Palawa). Aksara Palawa paling berpengaruh di Indonesia dan tulisan asli Indonesia berinduk pada aksara tersebut. Kerabat aksara Batak yang paling dekat

adalah aksara-aksara Nusantara (aksara-aksara turunan India yang terdapat di kepulauan Asia Tenggara), khususnya yang di Sumatera.



Gambar 2.1. Silsilah Aksara (Kozok, 2009)

Aksara Nusantara asli dapat dibagi atas lima kelompok :

1. Aksara Hanacaraka (Jawa, Sunda, Bali)
2. Surat Ulu (Kerinci, Rejang, Lampung, Lembak, Pasemah dan Serawi)
3. Surat Batak (Angkola-Mandailing, Toba, Simalungun, Pakpak Dairi, Karo)
4. Aksara Sulawesi (Bugis, Makasar dan Bima)
5. Aksara Filipina (Bisaya, Tagalog, Tagbanwa, Mangyan)

Aksara Batak diklasifikasikan sebagai abugida, yaitu paduan antara silabogram (sistem tulisan dengan satuan dasar berupa konsonan yang diikuti oleh sebuah vokal) dan abjad. Seperti aksara Batak dan aksara Nusantara lainnya,

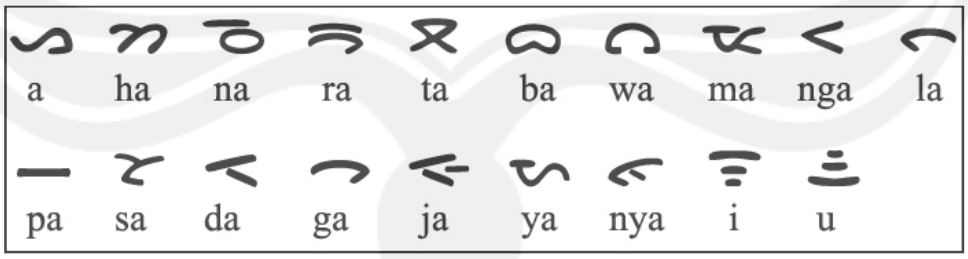
aksara Batak Toba terdiri atas dua perangkat huruf yang masing-masing di sebut *ina ni surat* dan *anak ni surat*. Urutan yang diketahui selama ini dan sering dipakai di sekolah adalah a-ha-na-ra-ta-ba-wa-i-ma-nga-la-pa-sa-da-ga-ja. Urutan ini adalah ciptaan baru dan tidak memiliki landasan tradisional. Urutan tersebut digunakan agar mudah untuk diingat dalam bentuk kalimat “aha na rata bawa i mangalapa sada gaja” yang artinya “apa yang hijau lelaki itu memotong seekor gajah”.

2.3.1.1. Ina Ni Surat

Ina ni surat (*ina* = ibu) terdiri dari huruf-huruf silabik dasar yang diakhiri bunyi /a/ (kecuali untuk huruf i dan u) seperti yang ditunjukkan tabel 2.1.

Tabel 2.1. Huruf-huruf *ina ni surat* dan variannya (Kozok, 2009; Simatupang, 2006)

a	a varian a	**ta**	t	**nga**	<	**da**	d	**nya**	[
ha/ka	h	**ba**	b	**la**	l	**ga**	g	**i**	I
na	n varian n	**wa**	w varian w	**pa**	p varian p	**ja**	j	**u**	U
ra	r	**ma**	m varian m	**sa**	s	**ya**	y		



Gambar 2.2. Huruf-huruf *Ina ni surat* (Font tradisional)

2.3.1.2. Anak Ni Surat

Bunyi /a/ pada *ina ni surat* dapat diubah dengan menambah nilai fonetisnya. Pengubah ini disebut diakritik. Diakritik dalam *anak ni surat* sebagai berikut :

1. Bunyi /e/ (pepet/keras) disebut ‘hatadingan’, dengan menambah garis kecil disebelah kiri atas *ina ni surat,* contoh :

   /pa/ p pe p

   /ba/ b be b

   /ga/ g ge g

2. Bunyi /ng/ disebut ‘paninggil’, dengan menambah garis kecil disebelah kanan atas *ina ni surat,* contoh :

   /pa/ p pang p

   /ba/ b bang b

   /ga/ g gang g

3. Bunyi /u/ disebut ‘haborotan’ disebelah bawah *ina ni surat,* contoh :

   /pa/ p pu P

   /ba/ b bu B

   /ga/ g gu G

4. Bunyi /i/ disebut ‘hauluan’ bentuk lingkaran kecil setelah *ina ni surat,* contoh:

   /pa/ p pi p i

   /ba/ b bi b i

   /ga/ g gi g i

5. Bunyi /o/ disebut ‘sihora’ atau ‘siala’ berupa tanda kali setelah *ina ni surat*, contoh:

   /pa/ p po p o

   /ba/ b bo b o

   /ga/ g go g o

6. Tanda mati untuk menghilangkan bunyi /a/ pada *ina ni surat* disebut ‘pangolat’, contoh:

   laha l h menjadi ‘lah’ l h \

   pasa p s menjadi ‘pas’ p s \

   mara m r menjadi ‘mar’ m r \

2.3.2. Pengenalan Pola

Pengenalan pola adalah salah satu cabang dari bidang kecerdasan buatan. Beberapa penulis (Liu *et all*, 2006; Fatta, 2009) mengutip beberapa definisi pengenalan pola dari beberapa peneliti sebelumnya, yaitu penentuan suatu objek fisik atau kejadian ke dalam salah satu atau beberapa kategori (Duda dan Hart, 1973), ilmu pengetahuan yang menitik-beratkan pada deskripsi dan klasifikasi (pengenalan) dari suatu pengukuran (Schalkoff, 1992). Secara garis besar dapat dirangkum bahwa pengenalan pola merupakan cabang kecerdasan buatan yang menitikberatkan pada metode pengklasifikasian objek ke dalam kelas-kelas tertentu untuk menyelesaikan masalah tertentu, dengan memetakan (menggambarkan sesuatu berdasarkan pengukuran kuantitatif) suatu fitur, yang merupakan ciri utama suatu objek (yang dinyatakan dalam sekumpulan bilangan-

bilangan) ke suatu kelas yang sesuai. Proses pemetaan ini menyangkut inferensi, baik secara eksplisit menggunakan statistik (misalnya dalam aturan Bayesian) maupun tak eksplisit dengan suatu jaringan keputusan (misalnya JST atau logika *fuzzy*) (Fatta, 2009). Sedangkan **pola** adalah suatu entitas yang terdefinisi (mungkin secara samar) dan dapat diidentifikasi serta diberi nama. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor dan matriks, contoh : sidik jari, raut wajah, gelombang suara, tulisan tangan dan lain sebagainya (Murni, 1992; Putra, 2010). Dalam pengenalan pola, data yang akan dikenali biasanya dalam bentuk citra atau gambar, akan tetapi ada pula yang berupa suara.

Pengenalan pola memainkan peran penting dalam berbagai bidang rekayasa, Beberapa contoh pemanfaatan pengenalan pola salah satunya untuk sistem keamanan, misalnya pengenalan suara yang digunakan sebagai kunci pada sistem rahasia, pengenalan sidik jari yang dipakai sebagai pengganti *password* atau *pin* untuk mengakses sistem komputer tertentu, pengenalan wajah untuk mengidentifikasi para buronan dengan melakukan *scanning* pada sejumlah besar data wajah para pelaku yang sudah masuk dalam basis data berdasarkan foto pelaku kejahatan tersebut dan sebagai pengganti *password* atau PIN. Pemanfaatan pengenalan pola lainnya adalah pengenalan tulisan tangan atau karakter untuk mengenali bentuk karakter secara terbatas, sehingga pemindahan isi suatu dokumen ke dalam komputer untuk diolah lebih lanjut dapat dilakukan secara otomatis tanpa proses pengetikan kembali dokumen tersebut. Contoh aplikasi tersebut adalah aplikasi perbankan menggunakan pengenalan tulisan untuk

membuktikan pelaku transaksi adalah orang yang benar-benar berhak. Selain itu pengenalan tulisan tangan dimanfaatkan untuk penyortiran surat secara otomatis. Form atau surat dipindai sehingga menghasilkan gambar digital yang diubah menjadi karakter-karakter yang akan disimpan ke dalam basis data. Di bidang kesehatan atau kedokteran pengenalan pola dimanfaatkan untuk pengenalan citra medis yang dirancang untuk membantu para dokter untuk diagnosis, contohnya citra *X-ray*, klasifikasi sel darah putih. Pemanfaatan pengenalan pola lainnya adalah di bidang militer misalnya untuk pengenalan objek yang dibidik secara otomatis, di bidang geografi untuk klasifikasi bebatuan dan topografi permukaan bumi, atau di bidang pertanian misalnya untuk mengetahui ragam vegetasi, jenis dan kondisi tanah (Liu *et all*, 2006; Fatta, 2009; Murni, 1992).

2.3.3. Pengenalan Tulisan Tangan

Menurut Plamondon dan Srihari (Wu dan Yu, 2008), pengenalan tulisan tangan adalah proses perubahan suatu bahasa yang dihadirkan dalam bentuk ruang melalui tulisan menjadi representasi simbolik.

Tahapan umum yang dilakukan pada sistem pengenalan tulisan tangan terdiri dari (Cheriet *et all*, 2007; Fatta, 2009; Putra, 2010) :

1. *Data acquisition* / pemerolehan data

   Tahapan ini adalah tahap pemerolehan data dari sensor (misal pada kamera) yang digunakan untuk menangkap objek dari dunia nyata dan selanjutnya diubah menjadi sinyal digital (sinyal yang terdiri dari sekumpulan bilangan) melalui proses digitalisasi. Terdapat dua metode utama yang

digunakan untuk pengenalan tulisan (yang biasa disebut *recognition system*), yaitu pengenalan tulisan secara *offline* dan *online*. Pada sistem pengenalan tulisan *online*, inputan tulisan bersifat temporer diperoleh secara langsung dari alat input digital. Contoh sistem pengenalan tulisan tangan *online* adalah pengenalan tulisan tangan pada PDA. Pada sistem pengenalan tulisan secara *offline*, input tulisan diperoleh dari teks tulisan yang di-*scan* terlebih dahulu atau dari kamera (Abu-Ain *et all*, 2011; Al-Alaoui *et all*, 2009; Leila dan Mohammed, 2007). Contoh pengenalan tulisan tangan *offline* adalah pembacaan alamat pada kartu pos atau surat, data-data yang ditulis pada form tertentu yang kemudian dibaca oleh OCR dan lainnya.

2. *Data preprocessing* / pemrosesan awal data

Pada tahapan ini informasi dari citra huruf ditonjolkan, derau/*noise* dan kompleksitas ciri diminimalisasi, ukuran dan bentuk huruf dinormalisasikan agar diperoleh akurasi yang tinggi untuk tahap selanjutnya.

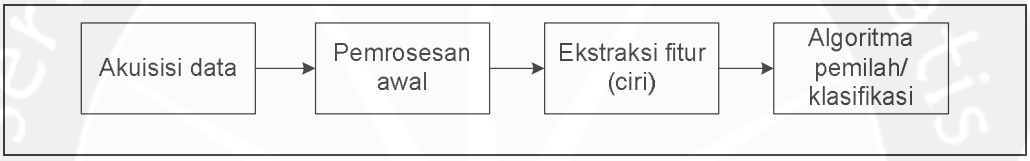
3. *Feature extraction* / ekstraksi ciri

Pada bagian ini terjadi ekstraksi ciri untuk mendapatkan karakteristik pembeda yang mewakili sifat utama dengan memisahkannya dari fitur yang tidak diperlukan untuk proses klasifikasi. Ekstraksi ciri biasanya diikuti oleh prosedur reduksi dimensi citra untuk mengurangi kerumitan komputasi pada tahap klasifikasi dan memungkinkan untuk meningkatkan akurasi.

4. *Data recognition (classification)* / pengenalan data (klasifikasi)

Tahapan ini berfungsi untuk mengelompokkan fitur ke dalam kelas yang sesuai dengan menggunakan algoritma klasifikasi tertentu. Hasil dari tahapan ini adalah klasifikasi dari objek yang ditangkap ke dalam kriteria-kriteria yang telah ditentukan. Output dari klasifikasi adalah sebuah kelas karakter yang unik atau sebuah daftar urut dari kelas-kelas dengan nilai kepercayaan masing-masing.

Tahapan pengenalan tulisan tangan digambarkan seperti gambar 2.3.



Gambar 2.3. Tahapan pengenalan tulisan tangan (Fatta, 2009)

2.3.4. Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan (JST) adalah sebuah sistem pengolahan informasi yang karakteristik kinerjanya menyerupai jaringan saraf biologis. Jaringan saraf tiruan telah banyak dikembangkan sebagai generalisasi model matematika dari pengertian manusia atau saraf biologi, berdasarkan pada asumsi-asumsi bahwa :

1. Pemrosesan informasi terjadi pada banyak elemen-elemen sederhana yang disebut saraf (*neuron*).
2. Sinyal-sinyal disampaikan antar saraf atas/pada jalur-jalur hubungan (*connection link*).

3. Setiap jalur hubungan mempunyai sebuah bobot hubungan (*associated weight*), yang mana di dalam jaringan saraf yang khas, ini menggandakan sinyal transmisi/pengiriman.
4. Setiap saraf menggunakan fungsi aktivasi (*activation function*), biasanya nonlinier, untuk jaringan inputnya (penjumlahan dari bobot sinyal input) untuk menentukan sinyal outputnya.

JST ditentukan oleh tiga hal :

1. Pola hubungan antar sarafnya (disebut arsitektur)
2. Metode penentuan bobot pada hubungan-hubungannya (disebut pembelajaran, pengetahuan atau algoritma)
3. Fungsi aktivasinya.

Sebuah jaringan saraf terdiri atas sejumlah besar elemen pemrosesan sederhana yang disebut saraf (*neuron*), unit (*units*), sel (cells), atau titik (*nodes*). Setiap saraf menerima sinyal dari lingkungannya atau jaringan saraf lainnya, dan mengirimkan sinyal tersebut ke saraf lain yang berhubungan, dengan memakai jalur komunikasi langsung, masing-masing disebut dengan bobot hubungan (Engelbrecht, 2007; Santoso, 2000). Bobot menunjukkan informasi yang telah digunakan oleh jaringan untuk memecahkan masalah.

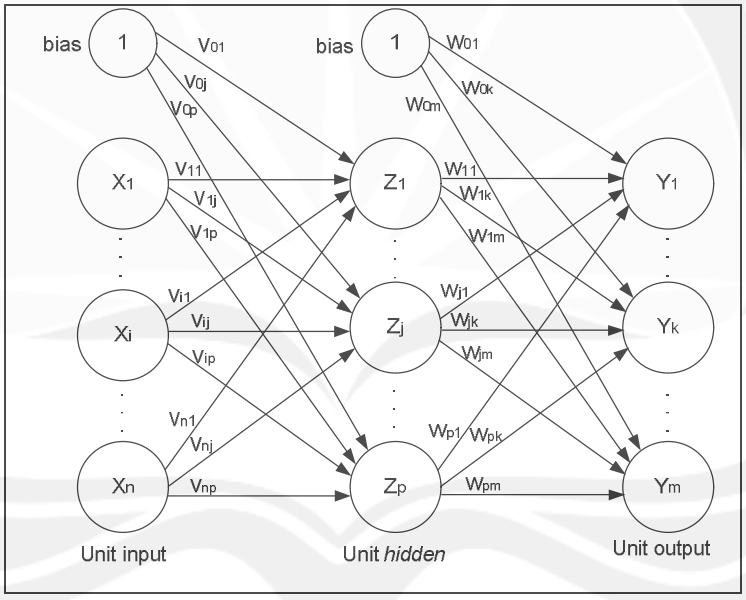
2.3.5. *Backpropagation*

JST yang menggunakan *Backpropagation* dapat dipahami dalam beberapa tingkatan. Pada satu tingkat *Backpropagation* merupakan kumpulan dari persamaan-persamaan vektor, pada tingkat lainnya dapat dipandang sebagai suatu

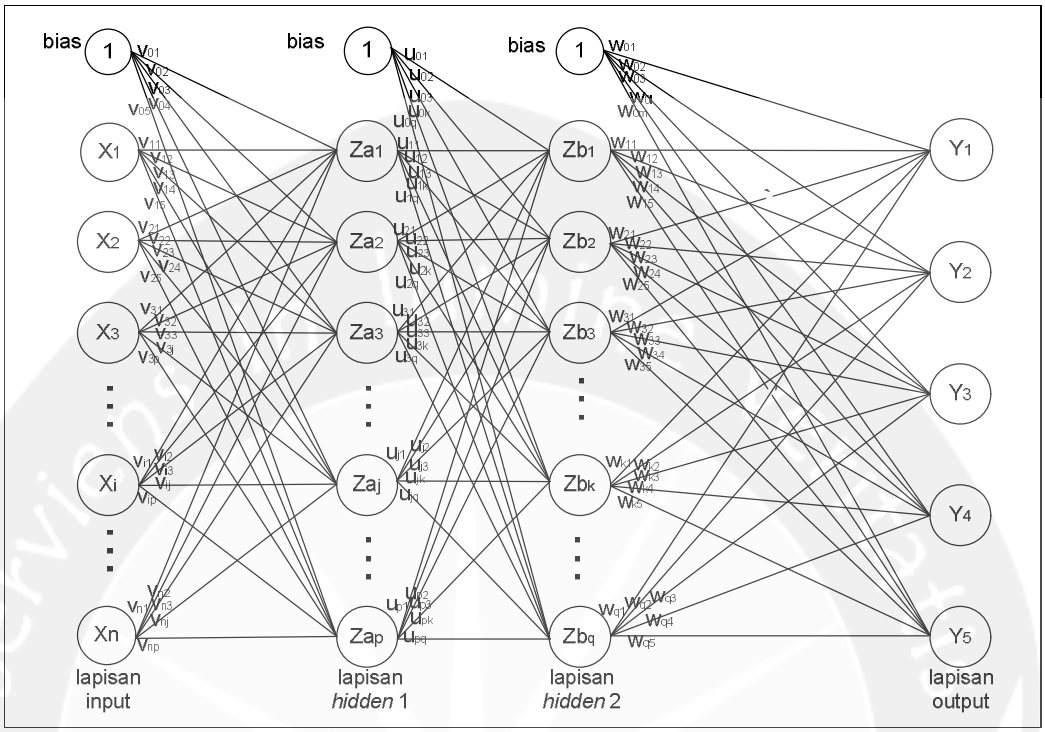
program komputer dan tingkat lainnya lagi dipandang sebagai suatu sistem berlapis dengan simpul-simpul yang saling berinteraksi. Data yang memasuki JST melalui input. Simpul-simpul pada lapisan input bersifat pasif, tidak melakukan perhitungan; masing-masing mengirimkan nilai data tersebut melalui bobot-bobot sambungan ke simpul-simpul yang tersembunyi (*hidden*). Semua simpul tersembunyi menerima data dari lapisan input, tetapi karena setiap simbol mempunyai paket bobot yang berbeda, maka hasil dari nilai paket-paket tersebut juga berbeda. Setiap simpul tersembunyi mengolah input-inputnya dan mengirimkan hasilnya ke lapisan output. Simpul-simpul output mempunyai satu paket bobot yang berbeda dan mengolah nilai input dan output. Hasil latihan adalah suatu paket nilai-nilai variabel, satu untuk setiap output. Simpul-simpul tersembunyi tidak mempunyai hubungan langsung dengan input dan output. Simpul-simpul tersembunyi dan simpul-simpul output mengolah input-input yang diterimanya dalam dua tahap. Masing-masing dari mereka mengalikan setiap input dengan bobot-bobotnya, menambahkan hasilnya untuk mendapatkan total penjumlahan dan kemudian melewatkan jumlahnya melalui suatu fungsi tertentu untuk mendapatkan hasil. Inti algoritma belajar *Backpropagation* terletak pada kemampuannya untuk mengubah nilai-nilai bobotnya untuk menanggapi adanya kesalahan (Santoso, 2000).

2.3.5.1. Arsitektur Jaringan

Backpropagation terdiri dari banyak lapisan (*multilayer neural network*), yaitu satu lapisan input mulai dari unit input 1 sampai n, lapisan tersembunyi (*hidden layer*) yang minimal berjumlah satu, mulai dari unit tersembunyi 1 sampai p dan lapisan output mulai dari unit output 1 sampai m. Nilai m, p, n adalah bilangan integer menurut arsitektur yang dirancang (Puspitaningrum, 2006). Arsitektur JST dengan 1 *hidden* layer ditunjukkan oleh gambar 2.4 dan arsitektur JST dengan 2 *hidden* layer ditunjukkan pada gambar 2.5.



Gambar 2.4. Arsitektur *Backpropagation* dengan satu *hidden layer* (Fausset, 1994)



Gambar 2.5. Contoh Arsitektur *Backpropagation* dengan Dua *Hidden Layer*

2.3.5.2. Fungsi Aktivasi

Fungsi aktivasi yang dipakai dalam algoritma *Backpropagation* harus memenuhi beberapa syarat yaitu: kontinu, dapat didiferensiasi dan secara monoton tidak menurun. Salah satu fungsi aktivasi yang sesuai adalah fungsi sigmoid biner dengan range (0,1) dengan rumus (Siang, 2009) :

$$f_1(x) = \frac{1}{1+\exp(-x)} \quad (2.1)$$

dengan turunannya:

$$f_1'(x) = f_1(\mathrm{x})[1 - f_1(\mathrm{x})] \quad (2.2)$$

Fungsi lainnya adalah fungsi sigmoid bipolar dengan range (-1,1) yang memiliki rumus.

$$f_2(x) = \frac{2}{1+\exp(-x)} - 1 \quad (2.3)$$

dengan turunannya:

$$f_2\,'(x) = \frac{1}{2}[1 + f_2(x)][1 - f_2(x)] \quad (2.4)$$

2.3.5.3. Algoritma Pelatihan

Algoritma pelatihan untuk *Backpropagation* 1 *hidden* layer adalah sebagai berikut (Fausett, 1994) :

Langkah 0 : inisialisasi bobot (harga acak kecil)

Langkah 1 : selama kondisi berhenti salah, kerjakan langkah 2-9

Langkah 2 : untuk setiap pasangan, lakukan langkah 3-8

Feedforward :

Langkah 3 : setiap unit input (X_i, i = 1,…,n), menerima sinyal masukan x_i, dan mengirimkan ke semua unit lapisan tersembunyi.

Langkah 4 : setiap lapisan tersembunyi (Z_j, j = 1,…,p), jumlahkan sinyal input bobotnya :

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad (2.5)$$

gunakan fungsi aktivasi untuk menghitung sinyal keluaran :

$$z_j = f(z\_in_j) \quad (2.6)$$

dan kirim sinyal ini ke semua unit keluaran.

Langkah 5 : setiap unit output (Yk, k = 1,…,m), jumlahkan sinyal input bobotnya,

$$y\_in_k = w_{ok} + \sum_{j=1}^{p} z_j w_{jk} \quad (2.7)$$

dan gunakan fungsi aktivasinya untuk menghitung sinyal keluaran

$$y_k = f(y\_in_k) \quad (2.8)$$

Kesalahan *Backpropagation* (tahap umpan balik):

Langkah 6 : setiap unit keluaran (Y_k, k = 1,…,m), menerima pola target yang berhubungan dengan pola pelatihan input, hitung informasi kesalahannya,

$$\delta_k = (t_k - y_k)f'(y\_in_k) \quad (2.9)$$

menghitung koreksi bobot

$$\Delta w_{jk} = \alpha\delta_k z_j \quad (2.10)$$

menghitung koreksi bias

$$\Delta w_{ok} = \alpha\delta_k \quad (2.11)$$

mengirim harga δ_k ke unit-unit lapisan bawah.

Langkah 7 : setiap unit tersembunyi (Z_j, j=1,…,p) jumlahkan input delta

$$\delta\_in_j = \sum_{j=1}^{m} \delta_k w_{jk} \quad (2.12)$$

kalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error*

$$\delta_j = \delta_{in_j} f'(z\_in_j), \quad (2.13)$$

menghitung koreksi bobot :

$$\Delta v_{ij} = \alpha\delta_j x_i \quad (2.14)$$

dan menghitung koreksi bias

$$\Delta v_{oj} = \alpha\delta_j \quad (2.15)$$

Perbarui bobot dan bias :

Langkah 8 : setiap unit keluaran (Y_k, k=1,…,m) memperbarui bias dan bobot (j=0,…,p) :

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (2.16)$$

Setiap unit tersembunyi (Zj, j=1,…,p) memperbaui bias dan bobot (i=0,…,n)

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad (2.17)$$

Langkah 9 : pengetesan kondisi berhenti.

JST *Backpropagation* dengan dua *hidden* layer memiliki algoritma pelatihan yang mengalami perubahan seperti berikut (Fausett, 1994).

Langkah 0 : inisialisasi bobot (harga acak kecil)

Langkah 1 : selama kondisi berhenti salah, kerjakan langkah 2-11

Langkah 2 : untuk setiap pasangan, lakukan langkah 3-10

Feedforward :

Langkah 3 : setiap unit input (X_i, i = 1,…,n), menerima sinyal masukan x_i, dan mengirimkan ke semua unit *hidden layer* 1.

Langkah 4 : setiap unit *hidden layer 1* (Za_j, j = 1,…,p), jumlahkan sinyal input bobotnya :

$$za\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad (2.18)$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran:

$$za_j = f(za\_in_j) \quad (2.19)$$

dan kirim sinyal ini ke semua unit keluaran.

Langkah 5 : setiap unit *hidden layer 2* (Zb_k, k = 1,…,q), jumlahkan sinyal input bobotnya,

$$zb\_in_k = w_{ok} + \sum_{j=1}^{p} za_j w_{jk} \quad (2.20)$$

dan gunakan fungsi aktivasinya untuk menghitung sinyal keluaran

$$zb_k = f(zb\_in_k) \quad (2.21)$$

Langkah 6 : setiap unit output (Yl, l = 1,…,m), jumlahkan sinyal input bobotnya,

$$y\_in_l = w_{ol} + \sum_{k=1}^{q} zb_k w_{kl} \quad (2.22)$$

dan gunakan fungsi aktivasinya untuk menghitung sinyal keluaran

$$y_l = f(y\_in_l) \quad (2.23)$$

Kesalahan *Backpropagation* (tahap umpan balik):

Langkah 7 : setiap unit keluaran (Y_l, l = 1,…,m), menerima pola target yang berhubungan dengan pola pelatihan input, hitung informasi kesalahannya,

$$\delta\_out_l = (t_l - y_l) f'(y\_in_l) \quad (2.24)$$

menghitung koreksi bobot

$$\Delta w_{kl} = \alpha . \delta\_out_l . zb_k \quad (2.25)$$

menghitung koreksi bias

$$\Delta w_{ol} = \alpha \delta\_out_l \quad (2.26)$$

mengirim harga δ_out_l ke unit-unit lapisan bawah.

Langkah 8 : setiap unit tersembunyi (Zb_k, k=1,…,q) jumlahkan input delta

$$\delta\_hid2\_in_k = \sum_{l=1}^{m} \delta\_out_l w_{kl} \quad (2.27)$$

kalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error*

$$\delta\_hid2_k = \delta\_hid2\_in_k . zb_k . (1 - zb_k) \quad (2.28)$$

menghitung koreksi bobot:

$$\Delta w_{jk} = \alpha . \delta\_hid2_k . za_j \quad (2.29)$$

dan menghitung koreksi bias:

$$\Delta w_{ok} = \alpha . \delta\_hid2_k \quad (2.30)$$

Langkah 9 : setiap unit tersembunyi (Za_j, j=1,…,p) jumlahkan input delta

$$\delta\_hid1\_in_j = \sum_{k=1}^{q} \delta\_hid2_k w_{jk} \quad (2.31)$$

kalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error*

$$\delta\_hid1_j = \delta\_hid1\_in_j . za_j . (1 - za_j) \quad (2.32)$$

menghitung koreksi bobot :

$$\Delta v_{ij} = \alpha . \delta\_hid1_j . x_i \quad (2.33)$$

dan menghitung koreksi bias:

$$\Delta v_{oj} = \alpha . \delta\_hid1_j \quad (2.34)$$

Perbarui bobot dan bias :

Langkah 10 : setiap unit keluaran (Y_l, l=1,…,m) memperbarui bias dan bobot (k=0,…,q) :

$$w_{kl}(baru) = w_{kl}(lama) + \Delta w_{kl} \quad (2.35)$$

Setiap unit tersembunyi (Zb_k, k = 1, …, q) memperbaui bias dan bobot (j=0,…,p)

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (2.36)$$

Setiap unit tersembunyi (Za_j, j=1,…,p) memperbaui bias dan bobot (i=0,…,n)

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad (2.37)$$

Langkah 11 : pengetesan kondisi berhenti.

2.3.5.4. Pembaharuan Bobot dengan Momentum

Penambahan parameter momentum dalam meng-*update* bobot seringkali bisa mempercepat proses pelatihan. Hal ini disebabkan karena momentum memaksa proses perubahan bobot terus bergerak sehingga tidak terperangkap dalam minimum-minimum lokal. Jika parameter momentum digunakan maka persamaan-persamaan peng-*update*-an bobot dengan langkah pelatihan t, dan t+1 untuk langkah pelatihan selanjutnya mengalami modifikasi sebagai berikut (Fausset, 1994) :

$$\Delta w_{jk}(t+1) = \alpha\, \delta_k\, z_j + \mu\, \Delta w_{jk}(t) \quad (2.38)$$

dan

$$\Delta v_{ij}(t+1) = \alpha\, \delta_j\, x_1 + \mu\, \Delta v_{ij}(t) \quad (2.39)$$

sehingga persamaan untuk menghitung bobot baru adalah sebagai berikut :

$$w_{jk}(t+1) = w_{jk}(t) + (\alpha\, \delta_k z_j + \mu.(w_{jk}(t) - w_{jk}(t-1)) \quad (2.40)$$

dan

$$v_{ij}(t+1) = v_{ij}(t) + (\alpha \delta_j x_i + \mu.(v_{ij}(t) - v_{ij}(t-1)) \quad (2.41)$$

dengan

w_{jk} (t), v_{ij}(t) = bobot awal pola kedua (hasil dari iterasi pola pertama).

w_{jk} (t−1), v_{ij}(t-1) = bobot awal pada iterasi pola pertama

μ = parameter momentum dalam range antara 0 sampai 1

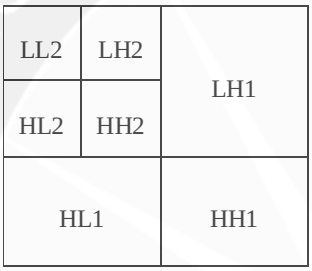
2.3.6. *Wavelet*

Dalam proses ekstraksi ciri dilakukan transformasi citra untuk mendapatkan informasi yang lebih jelas yang terkandung dalam citra tersebut. Transformasi atau alih ragam citra pada bagian ini adalah perubahan ruang (domain) citra ke domain lainnya. Melalui proses transformasi, citra dapat dinyatakan sebagai kombinasi linier dari sinyal dasar (*basic signals*) yang sering disebut dengan fungsi basis (*basis function*) (Putra, 2010).

Wavelet diartikan sebagai *small wave* atau gelombang singkat. Transformasi *Wavelet* akan mengkonversi suatu sinyal ke dalam sederetan *Wavelet*. Gelombang singkat tersebut merupakan fungsi yang terletak pada waktu berbeda. Transformasi *Wavelet* mampu memberikan informasi frekuensi yang muncul dan memberikan informasi tentang skala atau durasi atau waktu. *Wavelet* dapat digunakan untuk menganalisa suatu bentuk gelombang (sinyal) sebagai kombinasi dari waktu (skala) dan frekuensi (Putra, 2010).

Proses transformasi pada *Wavelet* dapat dicontohkan sebagai berikut. Citra yang semula ditransformasi dibagi (didekomposisi) menjadi empat sub-citra baru untuk menggantikannya. Setiap sub-citra berukuran ¼ kali dari citra asli. Tiga

sub-citra pada posisi kanan atas, kanan bawah dan kiri bawah akan tampak seperti versi kasar dari citra asli karena berisi komponen frekuensi tinggi dari citra asli. Sedangkan untuk sub-citra pada posisi kiri atas tampak seperti citra asli dan lebih halus, karena berisi komponen frekuensi rendah dari citra asli. Sub-citra pada bagian kiri atas (frekuensi rendah) tersebut dibagi lagi menjadi empat sub-citra baru. Proses diulang sesuai dengan level transformasi yang diinginkan. Untuk lebih jelasnya ditunjukkan pada gambar 2.6.



LH1, HL1, dan HH1 merupakan hasil dekomposisi level 1.

LL1 tidak diperlihatkan pada gambar karena langsung didekomposisi lagi menjadi LL2, LH2, HL2 dan HH2

Gambar 2.6. Dekomposisi Citra (Santoso, 2011)

Pada citra 2 dimensi, terdapat dua cara untuk mentransformasi atau mendekomposisi nilai-nilai pikselnya, yaitu dekomposisi standar dan tak standar. Keduanya diperoleh berdasarkan transformasi *Wavelet* 1 dimensi.

Dekomposisi standar menggunakan transformasi *Wavelet* 1 dimensi pada tiap baris citra dan kemudian pada tiap kolom. Dekomposisi tak standar diperoleh dengan mengkombinasikan pasangan transformasi baris dan transformasi kolom secara bergantian. Pada langkah pertama diterapkan transformasi *Wavelet* 1 dimensi pada baris, kemudian diterapkan alihragam *Wavelet* 1 dimensi pada kolom, proses tersebut diulang sesuai dengan level yang diinginkan.

2.3.6.1. Wavelet Haar

Wavelet Haar diperkenalkan oleh Alfred Haar pada tahun 1909. Untuk *Wavelet Haar* 1 dimensi, fungsi basis untuk ruang Vj disebut dengan fungsi penyekalaan. Basis sederhana pada Vj diberikan sebagai fungsi penyekalaan dan translasi sebagai berikut (Putra, 2010) :

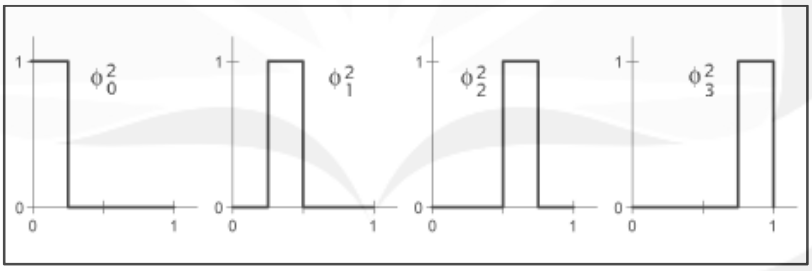
$$\emptyset_i^j(x) = \emptyset(2^j x - i) \quad i = 0,1,2,\ldots,2^j\text{-}1 \tag{2.42}$$

dengan

$$\emptyset(x) = \begin{cases} 1 & untuk\ 0 \leq x < 1 \\ 0 & untuk\ lainnya \end{cases}$$

$\emptyset(x)$ sering disebut dengan fungsi penyekalan (*scaling function*).

Sebagai contoh fungsi basis untuk ruang V^2 seperti telihat pada gambar 2.7 dibawah ini.



Gambar 2.7 Fungsi Basis Ruang V^2

Fungsi *wavelet* yang sesuai dengan fungsi penyekalaan diatas disebut dengan *wavelet Haar*, yang diberikan dengan persamaan:

$$\Psi_i^j(x) = \Psi(2^j x - i) \qquad i = 0,1,\ldots,2^j \tag{2.43}$$

dengan

$$\Psi(x) = \begin{cases} 1 & untuk\ 0 \leq x < ½ \\ -1 & untuk\ ½ \leq x < 1 \\ 0 & untuk\ lainnya \end{cases}$$

dan Ψ(x) sering disebut dengan fungsi *wavelet* (*wavelet function* atau *mother wavelet*).

2.3.7. C# dan *.NET Framework*

C# merupakan bahasa pemrograman berorientasi objek untuk *framework* Microsoft .NET. C# didasarkan pada bahasa pemrograman C++ yang memiliki kemiripan dengan beberapa bahasa pemrograman seperti Visual Basic, Java dan Delphi. C# memerlukan CLR (*Common Language Runtime*) untuk dapat dijalankan, yang berarti juga membutuhkan *.NET Framework*. Pada CLR tersedia *library-library* yang diperlukan oleh aplikasi-aplikasi yang dibuat, dimana berlangsung eksekusi, pengelolaan sumber daya, hingga penanganan *error* secara otomatis. Untuk dapat membuat proyek-proyek menggunakan bahasa pemrograman C# dan memanfaatkan teknologi .NET yang harus dipersiapkan adalah .NET *Framework* dan *Microsoft Visual Studio* (Komputer, 2008; Ferdiana, 2006).