

Penentuan Harga Jual Properti secara Otomatis menggunakan Metode Probabilistic Neural Network

Gregorius S. Budhi¹, Justinus Andjarwirawan², Alvin Poernomo³
^{1,2,3} Fakultas Teknologi Industri, Program Studi Teknik Informatika, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60441
Email: greg@petra.ac.id, justin@petra.ac.id

Abstrak

Wealth Aspiration, Inc. merupakan sebuah perusahaan di California yang bergerak di bidang properti *real estate*. Dalam menjalankan fungsinya, *Wealth Aspiration* seringkali membutuhkan estimasi harga jual suatu properti yang selama ini masih dilakukan secara manual. Proses ini memakan waktu yang lama guna menganalisa data transaksi dan data lain terkait. Dalam penelitian ini dikembangkan sebuah aplikasi data mining untuk estimasi harga jual suatu properti dengan membandingkan kemiripan spesifikasi rumah yang dituju dengan rumah-rumah di sekitarnya dalam radius 1 mil dan memiliki sejarah transaksi jual-beli kurang dari 6 bulan. Estimasi dilakukan dengan menggunakan metode *Probabilistic Neural Network* dan dikembangkan dengan menggunakan bahasa pemrograman *Java*. Hasil pengujian menunjukkan bahwa aplikasi memiliki tingkat akurasi yang baik, yaitu 82,76% dan juga waktu proses yang cepat. Tingkat akurasi belum optimal dikarenakan adanya *adjustment* yang membutuhkan peranan manual dan *setting* bobot atribut yang juga masih dilakukan secara manual.

Kata kunci: Properti, Data Mining, Klasifikasi, Estimasi Harga, Probabilistic Neural Network

Pendahuluan

Memperkirakan harga jual suatu properti dipengaruhi oleh data transaksi yang jumlahnya banyak dan selalu berubah seiring dengan berjalannya waktu. Oleh sebab itu bila estimasi ini dilakukan secara manual akan memakan waktu yang lama guna menganalisa data transaksi dan data lain terkait. *Wealth Aspiration, Inc.* adalah sebuah perusahaan perantara perdagangan dan penasehat di bidang properti *real estate* yang berlokasi di California. Dalam menjalankan fungsinya sebagai perantara perdagangan dan penasehat *real estate*, *Wealth Aspiration, Inc.* memerlukan estimasi harga jual properti secara cepat dan akurat. Estimasi ini akan dimanfaatkan oleh user yang ingin mengetahui perkiraan harga jual dari propertinya. Untuk menjawab kebutuhan perusahaan, pada penelitian ini dibuat sebuah aplikasi yang dapat melakukan estimasi otomatis harga dari properti dengan menggunakan metode Jaringan Saraf Tiruan *Probabilistic Neural Network (PNN)*.

Landasan Teori

Data Mining

Data Mining adalah pencarian pola yang menarik dari sejumlah besar data yang bisa

terdapat pada *database*, *data warehouse*, ataupun repositori informasi lainnya (Han, Kamber & Pei, 2012). Secara umum, *data mining* merupakan serangkaian proses yang dilakukan untuk menemukan informasi yang penting dari sejumlah besar data.

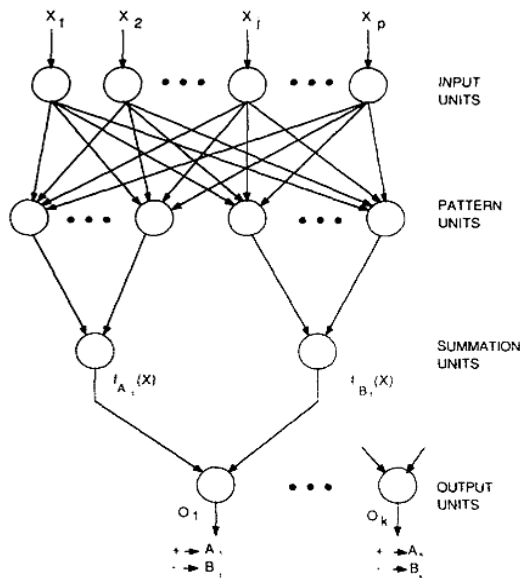
Probabilistic Neural Network

Probabilistic Neural Network (PNN) adalah sebuah metode jaringan syaraf tiruan yang menggunakan prinsip dari teori statistik yaitu *Bayesian Classification* untuk menggantikan prinsip heuristik yang digunakan oleh algoritma *Backpropagation* (Specht, 1990). Karena itulah *PNN* biasa digunakan untuk melakukan *pattern classification* (Fausett, 1994).

Arsitektur dari *PNN* terdiri dari 4 *layer*, yaitu *input layer*, *pattern layer*, *summation layer*, dan *decision layer / output layer* seperti dapat dilihat pada Gambar 1.

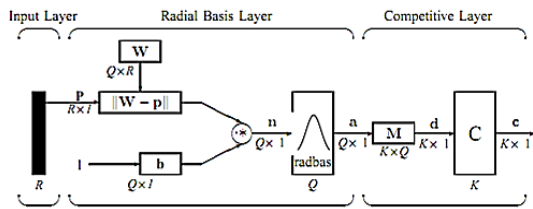
Input layer tidak melakukan perhitungan apapun, hanya mentransfer data *input* menuju tiap *neuron* pada *pattern layer*. Tiap *neuron* pada *pattern layer* akan melakukan perhitungan probabilitas (jarak) antara *input* dengan data yang tersimpan pada *neuron pattern layer*. Selanjutnya, *summation layer* akan menerima *input* dari tiap *neuron pattern layer* dan akan menjumlahkannya sehingga

akan didapat berapa kemungkinan suatu *input* x masuk ke dalam suatu kelompok t .



Gambar 1. Arsitektur PNN (Specht, 1990)

Dan terakhir, *output layer* akan menghasilkan hasil klasifikasi berdasarkan hasil dari *summation neuron* yang memiliki nilai yang paling besar. Dalam penelitian ini akan digunakan arsitektur seperti pada Gambar 2.



Gambar 2. Arsitektur PNN detail (Wu et al, 2007)

Berikut ini adalah algoritma *training* dan pengklasifikasian dari jaringan saraf tiruan *probabilistic* (Wu et al, 2007):

1. Melakukan inialisasi bobot awal (*radial basis layer weight*) W yang didapat dari hasil transpose dari matriks $R \times Q$ di mana R adalah dimensi input dan Q adalah jumlah *data training*.
2. Melakukan inialisasi konstanta *spread* (s) dari PNN.
3. Melakukan inialisasi bobot bias (b).

$$b = \frac{\sqrt{-\ln 0.5}}{s} \quad \text{Pers. 1}$$

4. Melakukan inialisasi bobot akhir (*competitive layer weight*) M yang merupakan matriks berukuran $K \times Q$ di mana K adalah jumlah kelompok hasil klasifikasi dan Q adalah jumlah data training. Baris ke- i pada matriks M mewakili

data training ke- i dan pada kolom ke- j nilai matriks akan bernilai 1 bila *data training* masuk ke dalam kelompok tersebut, bila tidak maka akan bernilai 0.

5. Menghitung jarak antara vektor data input P dengan vektor tiap baris pada bobot awal W (*Euclidean Distance* antara vektor P dengan vektor W_i) sehingga menghasilkan matriks jarak $\|W - P\|$ yang berdimensi $Q \times 1$.
 6. Menghitung nilai aktivasi dari jarak antara bobot awal dengan data input (vektor a), dengan menggunakan fungsi *radbas*(\cdot).
- $$\text{radbas}(n) = e^{-n^2} \quad \text{Pers. 2}$$
- $$n = \|\|W - P\|\| * b \quad \text{Pers. 3}$$
7. Mengalikan vektor a dengan matriks M sehingga menghasilkan vektor *output* d .
 8. Mencari output dari PNN dengan *competitive function* C di mana *competitive function* ini akan mencari nilai terbesar dalam vektor d . Index dari nilai terbesar inilah yang menandakan hasil klasifikasi data input P .
 9. Simpan bobot awal, *spread*, dan bobot akhir.

Min-Max Normalization

Pada algoritma *Probabilistic Neural Network*, *input* yang diterima berupa bilangan dengan *range* antara 0 sampai dengan 1. Karena itu, diperlukan sebuah proses terlebih dahulu untuk menyesuaikan data yang sesungguhnya dengan *range* tersebut.

Proses yang digunakan adalah *min-max normalization*, yaitu metode normalisasi yang mentransformasi data secara linear terhadap sebuah *range* yang baru (Han, Kamber & Pei, 2012). Jika min_A dan max_A masing-masing adalah nilai minimum dan maksimum dari sebuah atribut A , maka *min-max normalization* memetakan sebuah nilai v dari atribut A menjadi nilai v' dalam *range* $[new_min_A, new_max_A]$ melalui persamaan:

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A \quad \text{Pers. 4}$$

Partisi Natural

Metode Partisi Natural disebut juga sebagai 3-4-5 rule. Metode ini digunakan untuk mensegmentasi data numerik menjadi beberapa interval yang relatif sama (*uniform*) dan mudah dipahami. Aturan metode Partisi

Natural adalah sebagai berikut (Han, Kamber & Pei, 2012):

1. Jika interval mengandung 3, 6, 7, atau 9 nilai yang berbeda pada *most significant digit*, maka partisi *range* menjadi 3 interval (3 *equal-width* interval untuk 3, 6, dan 9; dan 3 interval dalam kelompok of 2-3-2 untuk 7).
2. Jika mengandung 2, 4, atau 8 nilai yang berbeda pada *most significant digit*, maka partisi *range* menjadi 4 *equal-width* interval.
3. Jika mengandung 1, 5, atau 10 nilai yang berbeda pada the *most significant digit*, maka partisi *range* menjadi 5 *equal-width* interval.

Metode ini dapat diterapkan dalam setiap interval secara rekursif sehingga tercipta sebuah konsep hierarki terhadap suatu data numerik.

Tahapan-tahapan dalam metode Partisi Natural adalah sebagai berikut:

1. Hitung nilai minimal, nilai maksimal, nilai *low* (5 *persentil*), dan nilai *high* (95 *persentil*) dari suatu atribut data numerik.
2. Cari nilai *most significant digit* antara nilai *low* dan *high*, kemudian bulatkan nilai *low* dan *high* ke digit *most significant digit* sehingga menjadi low^1 dan $high^1$.
3. Partisi interval *low-high* sesuai 3-4-5 *rule*. Hasil partisi ini yang akan menjadi hierarki paling atas.
4. Lakukan pengecekan terhadap nilai low^1 dan *min*. Jika *min* berada di dalam interval low^1 ($low^1 < min$), maka batas interval low^1 dapat digantikan oleh nilai *min*. Jika *min* berada di luar interval low^1 ($min < low^1$), maka dibuat sebuah interval baru dengan batas interval *min* dan low^1 . Lakukan hal yang sama terhadap nilai $high^1$ dan *max*.
5. Setiap interval dapat dilakukan partisi secara rekursif hingga terbentuk konsep hierarki yang sesuai.

Persamaan Haversine

Dengan menganggap bumi sebuah bola dengan radius R dan lokasi 2 titik pada permukaan bumi disebut *longitude* lon1 dan *latitude* lat1 serta lon2, lat2 maka perhitungan jarak antara 2 titik tersebut menurut persamaan *Haversine* adalah sebagai berikut (R. W. Sinnott, 1984):

$$dlon = lon_2 - lon_1 \quad \text{Pers. 5}$$

$$dlat = lat_2 - lat_1 \quad \text{Pers. 6}$$

$$a = \left(\sin\left(\frac{dlat}{2}\right) \right)^2 +$$

$$\cos(lat_1) \cos(lat_2) \left(\sin\left(\frac{dlon}{2}\right) \right)^2 \quad \text{Pers. 7}$$

$$c = 2 * \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \quad \text{Pers. 8}$$

$$d = R * c \quad \text{Pers. 9}$$

Pada penelitian ini persamaan *Haversine* digunakan untuk menggantikan perhitungan jarak antar 2 titik yang umum dipakai, yaitu rumus *Pythagoras*.

Metode Penelitian

Berikut adalah langkah-langkah yang dilakukan selama penelitian:

1. Analisis permasalahan estimasi harga rumah pada *Wealth Aspiration, Inc*.
2. Studi literatur tentang kemungkinan metode yang digunakan untuk memecahkan masalah, yaitu: *Data Mining*, klasifikasi data, *Probabilistic Neural Network*, persamaan *Haversine*, dan lain sebagainya.
3. Pembuatan software yang dibagi menjadi dua bagian besar, yaitu: bagian '*Data Preprocessing*' serta bagian '*Training dan Estimasi Harga Jual Rumah dengan PNN*'.
4. Pengujian software dari sisi akurasi estimasi dan kecepatan proses.
5. Pengambilan simpulan penelitian.
6. Dokumentasi dan publikasi hasil penelitian.

Data Spesifikasi Rumah yang digunakan untuk Estimasi Harga Jual

Spesifikasi rumah yang digunakan dalam aplikasi ini dibagi menjadi 2 bagian utama, yaitu atribut *mandatory* dan atribut *optional*. Atribut *mandatory* merupakan atribut / spesifikasi rumah yang wajib diikutsertakan dalam perhitungan, sedangkan atribut *optional* merupakan atribut / spesifikasi rumah yang tidak wajib diikutsertakan dalam perhitungan dan pemilihannya bergantung pada *professional appraiser / developer* yang bersangkutan.

Atribut-atribut *mandatory* yang dipakai pada aplikasi mencakup:

- *Address*: alamat suatu rumah yang terbagi menjadi:
 - ✓ *Number*: nomor rumah
 - ✓ *Fraction*: nomor yang menunjukkan letak rumah pada suatu jalan
 - ✓ *DirPrefix*: arah depan jalan
 - ✓ *StreetName*: nama jalan

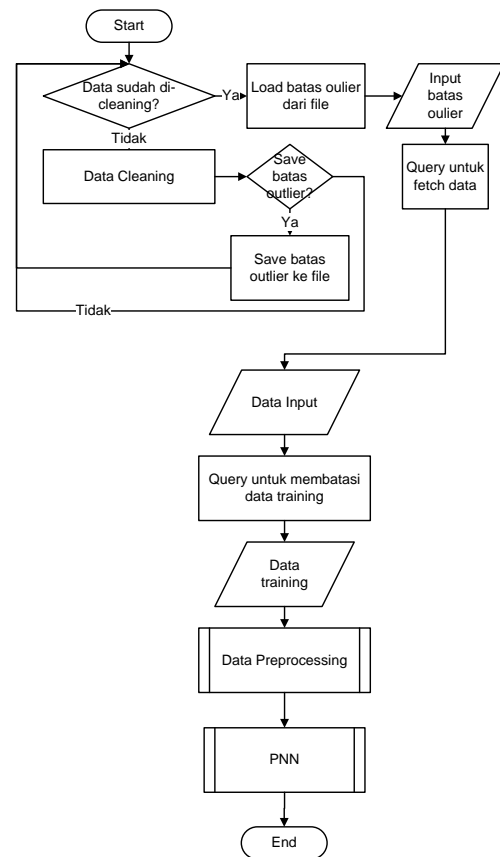
- ✓ *StreetType*: tipe jalan
 - ✓ *DirSuffix*: arah belakang jalan
 - ✓ *City*: nama kota
 - ✓ *State*: nama negara bagian
 - ✓ *ZipCode*: kode pos wilayah
 - *Bedrooms*: jumlah kamar tidur
 - *Bathrooms*: jumlah kamar mandi
 - *Building Sq_Ft*: luas bangunan dalam satuan *square feet*
 - *YearBuilt*: tahun pembangunan rumah.
- Sementara itu atribut-atribut *optional* yang dipakai pada aplikasi mencakup:
- *Stories*: jumlah tingkat/ lantai rumah
 - *Fullbaths*: jumlah fullbath (kamar mandi dengan toilet, wastafel, dan *shower*)
 - *FirePlaces*: jumlah pemanas api
 - *HeatType*: tipe pemanas/pendingin ruangan
 - *Condition*: tipe kondisi rumah.
- Data spesifikasi rumah kemudian direlasikan dengan data transaksi dari sebuah rumah guna melengkapi atribut-atributnya dengan yang ada pada data transaksi yaitu:
- *RecordingDate*: tanggal pencatatan dokumen transaksi
 - *Saleprice*: harga jual properti
 - *PersonalPropertyPrice*: harga barang-barang pribadi pemilik lama yang termasuk dalam harga jual
 - *PropertyType*: tipe properti
 - *PrincipalUse*: kegunaan utama suatu properti
 - *SaleInstrument*: tipe dokumen yang dipakai dalam proses transaksi
 - *SaleReason*: alasan penjualan suatu properti
 - *PropertyClass*: kategori suatu properti
 - *SaleWarning*: peringatan / kejanggalan transaksi.

Pada aplikasi, harga jual properti yang dipakai adalah harga jual bersih, yaitu harga properti dengan tidak termasuk harga barang-barang pribadi dari pemilik lama.

Data-data yang dipakai saat akan dilakukan estimasi harga jual sebuah rumah bukanlah semua data yang tersedia pada server namun dibatasi pada data-data rumah lain yang berjarak n mil dari rumah yang akan diestimasi (Diukur menggunakan persamaan *Haversine*, dan default n adalah 1 mil). Selain itu proses juga dibatasi hanya menggunakan data rumah yang mengalami transaksi jual-beli selama m bulan terakhir dari tanggal sekarang (default m adalah 6 bulan).

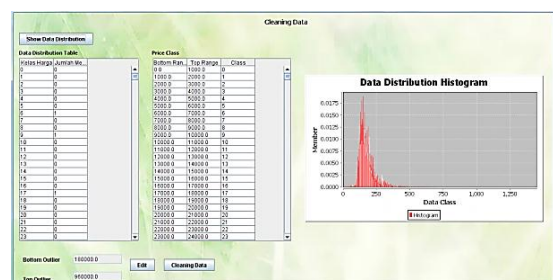
Desain dan Pengujian Software

Pada Gambar 3 dapat dilihat *flowchart* dari disain software.



Gambar 3. Flowchart dari disain software

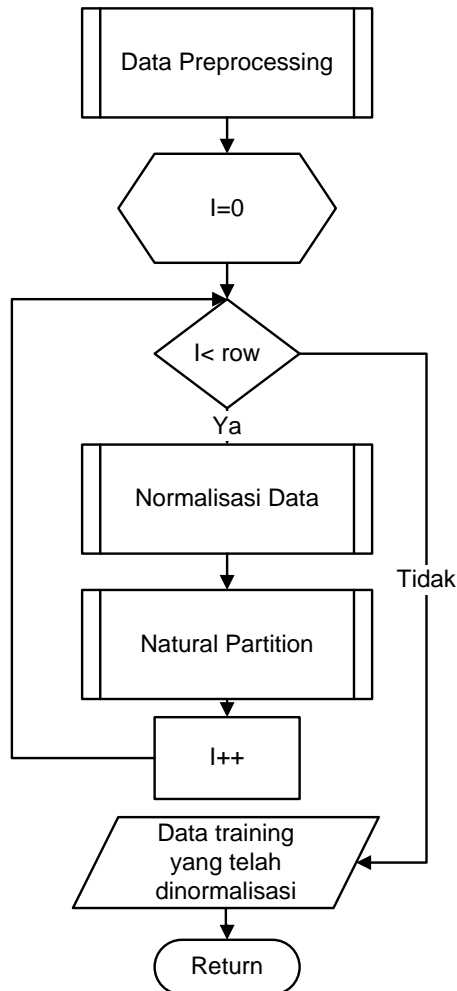
Proses *cleaning data* dalam aplikasi ini merupakan proses menentukan batas *outlier* dari data di mana data yang berada di luar batas *outlier* ini dianggap *noise* dan tidak diikutsertakan dalam proses selanjutnya. Batas *outlier* ditentukan dari perhitungan interkuartil. Selain itu, *user* juga dapat melakukan *edit/adjust* pada kedua batas *outlier* ini. Nilai batas *outlier* akan disimpan pada sebuah *file* sehingga *cleaning data* tidak perlu dilakukan berulang kali apabila aplikasi dijalankan kembali. Untuk menghitung jarak 2 lokasi Pada Gambar 4 dapat dilihat antarmuka dari proses *cleaning data*.



Gambar 4. Antarmuka proses *cleaning data*

Data Preprocessing

Data preprocessing terdiri dari 2 proses utama, yaitu metode normalisasi dan metode partisi natural partitioning. Gambaran umum cara kerja proses data preprocessing dapat dilihat pada Gambar 5. Sementara antarmuka proses ini dapat dilihat pada Gambar 6.



Gambar 5. Flowchart proses data preprocessing

Gambar 6. Antarmuka proses data preprocessing

Ada dua proses pada data preprocessing, yaitu:

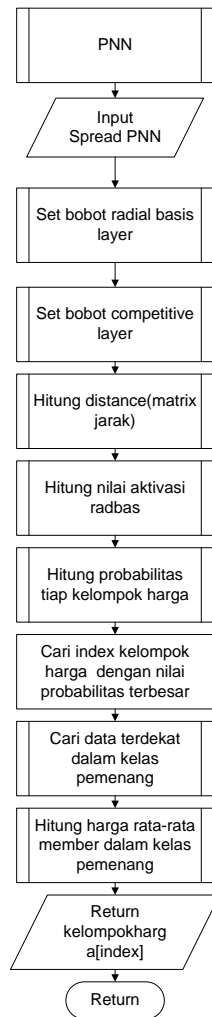
- **Normalisasi Data:** Data yang dapat diproses oleh metode PNN adalah data numerik dengan range 0 sampai dengan 1. Oleh sebab itu, perlu dilakukan normalisasi

data untuk data *input* dan data *training* yang sebelumnya memiliki *range* yang bermacam-macam. Normalisasi data dilakukan dengan menggunakan metode *Min-Max Normalization*.

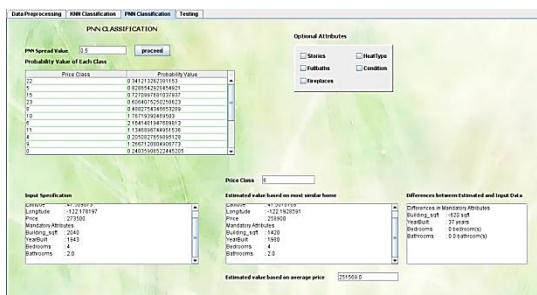
- **Partisi Natural:** Pada penelitian ini, secara khusus *natural partitioning* digunakan untuk mensegmentasi atribut harga pada data *input* dan data *training* menjadi beberapa kelas harga dengan *range* yang relatif *uniform*. Tujuannya agar harga rumah yang bervariasi dan tidak sama satu dengan lainnya dapat di generalisasi menjadi *range-range* kelompok harga rumah.

Proses Training dan Estimasi Harga Jual Rumah dengan PNN

Secara umum, proses *training PNN* hanyalah proses inialisasi *weight* awal dan *weight* akhir. *Weight* awal yang berupa *matrixW* merupakan *transpos* dari *matrix* data *training* itu sendiri, sedangkan *weight* akhir yang berupa *matrixM* merupakan *matrix* kelas/kelompok harga. Proses estimasi harga jual terdiri dari beberapa tahap, yaitu menghitung *matrix* jarak yang diperoleh dari perkalian *matrix* data *input* dengan *matrixW* yang kemudian dilakukan proses perhitungan nilai aktivasi dengan fungsi *radbas*. Hasil ini kemudian dikalikan dengan *matrixM* sehingga diperoleh nilai probabilitas data *input* dengan masing-masing kelas. Kelas dengan nilai probabilitas tertinggi nantinya akan digunakan untuk melakukan estimasi harga jual rumah. Gambaran umum cara kerja proses *training* dan estimasi harga jual rumah dengan metode PNN dapat dilihat pada Gambar 7. Antarmuka proses dapat dilihat di Gambar 8.



Gambar 7. Flowchart proses training dan estimasi harga rumah dengan PNN

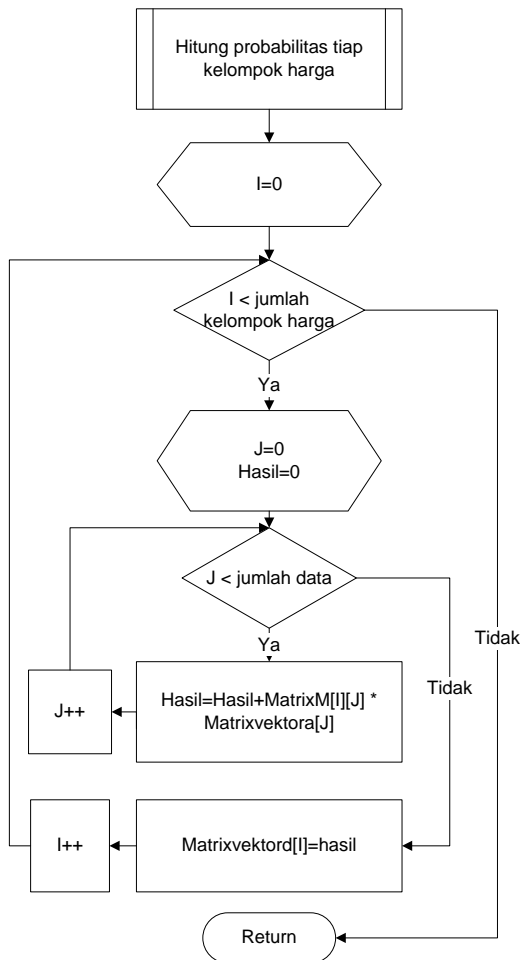


Gambar 8. Antarmuka proses training dan estimasi harga rumah dengan PNN

Berikut adalah pembahasan tiap-tiap proses didalam PNN.

- **Set Bobot Radial Basis Layer:** Bobot *radial basis layer* atau bobot awal merupakan sebuah matrix dengan ukuran $Q \times R$ di mana Q adalah jumlah data *training* dan R adalah jumlah atribut data. *Matrix* ini kemudian diisi sesuai dengan data *training* yang telah dinormalisasi sebelumnya. Inisialisasi bobot bias juga dilakukan proses ini.

- **Set Bobot Competitive Layer:** Bobot *competitive layer* atau bobot akhir merupakan sebuah *matrix* dengan ukuran $K \times Q$ di mana K adalah jumlah kelompok klasifikasi pada data *training* dan Q adalah jumlah data *training*. *Matrix* ini kemudian diisi sesuai dengan kelompok klasifikasi masing-masing data *training* dengan angka 1 dan sisanya 0.
- **Hitung Distance:** Perhitungan *distance* yang merupakan perkalian antara *matrix* W dengan *matrix* data *input* menggunakan persamaan *Euclidean Distance*. Hasil perkalian kedua *matrix* ini menghasilkan *matrix* jarak yang berukuran $Q \times 1$.
- **Hitung Nilai Aktivasi Radbas:** Perhitungan nilai aktivasi yang menggunakan fungsi *radbas* dilakukan pada setiap *index matrix* jarak yang kemudian dinamakan *matrix* vektor a .
- **Hitung Probabilitas Tiap Kelompok Harga:** Perhitungan nilai probabilitas tiap kelompok harga dilakukan dengan melakukan perkalian antara *matrix* M dengan *matrix* vektor a sehingga menghasilkan *matrix* vektor d yang berukuran $K \times 1$. Pada tiap-tiap *index matrix* vektor d inilah dapat dilihat nilai probabilitas data *input* terhadap masing-masing kelas/kelompok data *training*. Gambaran detil cara kerja proses perhitungan probabilitas tiap kelompok harga dapat dilihat pada Gambar 9.

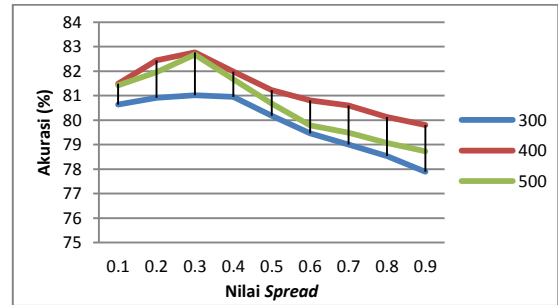


Gambar 9. Flowchart proses perhitungan probabilitas tiap kelompok harga

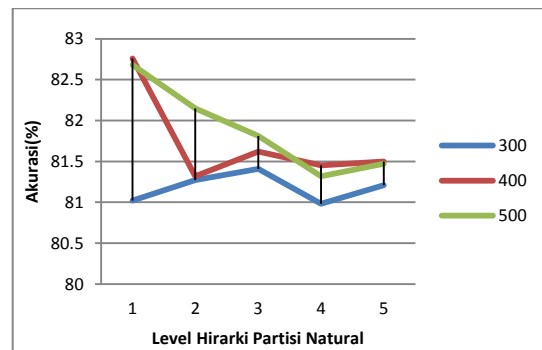
- Cari Data Terdekat dalam Kelompok Harga dengan Probabilitas Tertinggi:** Setelah memperoleh kelompok harga dengan probabilitas tertinggi, langkah berikutnya adalah mencari data terdekat dengan rumus perhitungan jarak *Euclidean*. Tujuannya adalah untuk memperoleh data yang paling mirip dengan data *input*, namun termasuk dalam kelompok harga dengan probabilitas tertinggi.
- Hitung Harga Rata-Rata Semua Member dalam Kelompok Harga dengan Probabilitas Tertinggi:** *Output* berikutnya yang ditampilkan ke *user* adalah harga rata-rata semua *member* dalam kelompok harga dengan probabilitas tertinggi. Tujuannya supaya *user* dapat membandingkan antara estimasi berdasarkan harga rata-rata dengan estimasi berdasarkan harga rumah yang paling mirip.

Pengujian

Pengujian akurasi dilakukan sebanyak tiga macam yaitu melihat pengaruh nilai *spread*, banyak sample data dan banyak level hirarki partisi natural terhadap akurasi dari estimasi harga jual rumah dibandingkan dengan harga riil dari rumah tersebut. Pada Gambar 10 dan 11 dapat dilihat hasil mengujian.



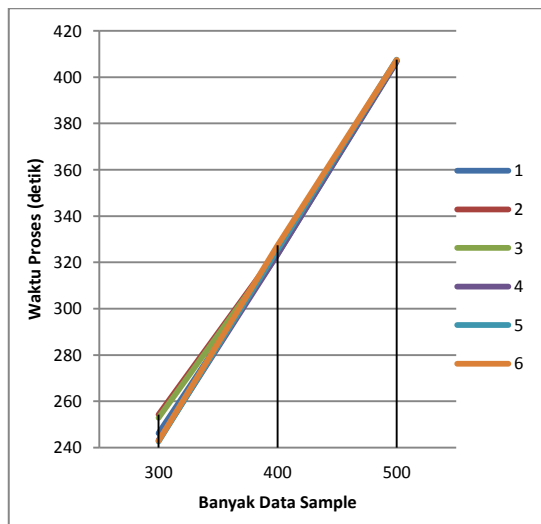
Gambar 10. Pengaruh nilai *spread* dan jumlah *sample* terhadap akurasi hasil estimasi



Gambar 11. Pengaruh banyak level hirarki partisi natural terhadap akurasi hasil estimasi

Dari hasil pengujian pada Gambar 11 dan Gambar 12 dapat disimpulkan bahwa nilai *spread* terbaik adalah 0.3, banyak sample terbaik adalah 400 sample data dan level hirarki partisi natural terbaik adalah level 1. Dengan menggunakan 3 parameter tersebut didapat hasil akurasi tertinggi yaitu 82.76%.

Untuk pengujian kecepatan proses dapat dilihat pada Gambar 13. Pada uji kecepatan proses ini, disetiap jumlah data sample yang sama dilakukan 6 kali pengujian dengan *setting* berbeda. Dari hasil pengujian ini dapat dilihat bahwa jumlah data memiliki pengaruh terbanyak pada kenaikan waktu proses, sementara perubahan parameter lain tidak. Semakin banyak jumlah data yang diproses semakin lama waktu yang dibutuhkan. Waktu proses ini dapat disebut cepat karena pada kondisi nyata saat pengaplikasian *software* ini jumlah data yang diproses untuk sekali estimasi jarang sekali lebih dari 100 data.



Gambar 12. Pengujian waktu proses PNN

Simpulan

Hasil pengujian menunjukkan bahwa *software* estimasi harga rumah yang dikembangkan dengan metode PNN ini memiliki tingkat akurasi yang baik, yaitu 82,76% dan juga waktu proses yang cepat. Tingkat akurasi belum optimal dikarenakan adanya *adjustment* yang membutuhkan peranan manual dan *setting* bobot atribut yang masih dilakukan secara manual. Dengan melakukan pengujian – pengujian lebih lanjut untuk *adjustment* dan juga *setting* bobot atribut yang benar akan dimungkinkan peningkatan akurasi dari estimasi harga rumah yang dihasilkan.

Daftar Pustaka

- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall International Inc.
- Han, J., Kamber, M. and Pei, J. (2012). *Data mining: Concepts and techniques (3rd ed.)*. Waltham USA: Morgan Kaufmann Publishers.
- Sinnott, R. W. (1984). Virtues of haversine. *Sky and Telescope*, vol. 68, no. 2, p. 159.
- Specht, D. F. (1990). Probabilistic Neural Network. *Neural Networks vol.3 no.1 pp.109-118*.
- Wu, S. G., Bao, S. F., Xu, E. Y., Wang, Y. X., Chang, Y. F., and Xiang, Q. L. (2007). A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. *2007 IEEE Int. Symposium on Signal Processing and Information Technology*. Pp. 11-16.