UNIVERSITAS INDONESIA

KARAKTERISASI JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* FUNGSI *ERROR* KUADRATIS DAN *CROSS-ENTROPY* DENGAN MENGGUNAKAN NILAI RATA-RATA PADA PERBAIKAN LEBAR DATA

SKRIPSI

RENALDI KRISSALAM
0806455433

FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JULI 2012

UNIVERSITAS INDONESIA

KARAKTERISASI JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* FUNGSI *ERROR* KUADRATIS DAN *CROSS-ENTROPY* DENGAN MENGGUNAKAN NILAI RATA-RATA PADA PERBAIKAN LEBAR DATA

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

RENALDI KRISSALAM
0806455433

FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JULI 2012

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama	**: Renaldi Krissalam**
NPM	**: 0806455433**
Tanda Tangan	**:**
Tanggal	**: 9 Juli 2012**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama	:	Renaldi Krissalam
NPM	:	0806455433
Program Studi	:	Teknik Elektro
Judul Skripsi	:	Karakterisasi Jaringan Saraf Tiruan *Radial Basis Function* Fungsi *Error* Kuadratis dan *Cross-Entropy* dengan Menggunakan Nilai Rata-Rata pada Perbaikan Lebar Data

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing :

Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng. ( )

Penguji 1 :

Ir.Wahidin Wahab M.Sc,Ph.D ( )

Penguji 2 :

Ir. Aries Subiantoro M.SEE ( )

Ditetapkan di : Depok

Tanggal : Juli 2012

UCAPAN TERIMA KASIH

Puji dan syukur kepada Tuhan Yang Maha Esa, atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Laporan Skripsi ini. Penulisan laporan ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik dan sebagai syarat untuk memenuhi mata kuliah Skripsi di Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari awal sampai akhir penyusunan buku laporan ini, sangatlah sulit bagi penulis untuk menyelesaikannya. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Prof. Dr.Eng. Drs. Benyamin Kusumoputro, M.Eng., selaku dosen pembimbing skripsi yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan laporan skripsi ini.
2. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral.
3. Rosandi Prarizki, Novia R. Putri, Musnida Ulya, dan Wisnu Indrajit yang telah membantu penulis dalam membuat simulasi dan mengaplikasikannya secara langsung.
4. Teman-teman Departemen Teknik Elektro angkatan 2008 dan pihak-pihak lain yang tidak dapat penulis sebutkan satu per satu yang telah memberi semangat dan mengingatkan penulis untuk menjadi lebih baik.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 9 Juli 2012

Renaldi Krissalam

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama	: Renaldi Krissalam
NPM	: 0806455433
Program Studi	: Teknik Elektro
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

"KARAKTERISASI JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN FUGNSI *ERROR* KUADRATIS DAN *CROSS-ENTROPY* DENGAN MENGGUNAKAN NILAI RATA-RATA PADA PERBAIKAN LEBAR DATA"

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 9 Juli 2012

Yang menyatakan,

(Renaldi Krissalam)

ABSTRAK

Nama : Renaldi Krissalam

Program Studi : Teknik Elektro

Judul : Karakterisasi Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error* Kuadratis dan *Cross-Entropy* dengan Menggunakan Nilai Rata-Rata pada Perbaikan Lebar Data

Pada masa sekarang ini perkembangan teknologi cenderung memiliki kemampuan untuk berpikir dan mengambil keputusan layaknya manusia. Salah satu dari banyak metode untuk mengembangkan teknologi yang cerdas adalah dengan menggunakan Jaringan Saraf Tiruan *Radial Basis Function*. Penelitian ini membandingkan antara Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error* Kuadratis dan *Cross-Entropy* dalam mengenal empat set data dari "*UCI Repository of Machine Learning Database*" dan satu set data uranium dari BATAN. Selain itu, kedua jaringan tersebut dibandingkan dengan Jaringan Saraf Tiruan *Backpropagation*. Berdasarkan hasil percobaan dapat dilihat bahwa algoritma *Radial Basis Function* lebih sederhana dan memiliki waktu komputasi yang lebih cepat dibandingkan dengan algoritma *Backpropagation*.

Kata kunci :

Teknologi cerdas, Jaringan Saraf Tiruan, *Radial Basis Function*

ABSTRACT

Name : Renaldi Krissalam

Study Program : Electrical Engineering

Title : Characterization of Radial Basis Function Neural Networks with Mean Square Error Function and Cross-Entropy Error Function by Using the Average Value on Improved Data Width

At the present, technological developments tend to have the ability to think and making decisions like human beings. One of the many methods to develop intelligent technologies is to use Radial Basis Function Neural Networks. This study compares the Radial Basis Functions Neural Networks with Mean Square Error Function and Cross-Entropy Error Function in identifying four sets of data from the "UCI Repository of Machine Learning Databases" and a set of data uranium from BATAN. In addition, both networks are compared with Backpropagation Neural Networks. Based on the results of the study, it is shown that Radial Basis Functions algorithm has simpler and faster computational capability compared to Backpropagation algorithm.

Key words:

Smart technology, Neural Networks, Radial Basis Function

DAFTAR ISI

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Di masa sekarang ini, perkembangan teknologi sangat maju dan pemakaian akan teknologi tersebut sangat luas serta tidak dapat dipisahkan dari kehidupan manusia. Perkembangan yang terjadi pada zaman sekarang ini cenderung untuk mengembangkan teknologi yang cerdas dengan memiliki kemampuan untuk berpikir dan mengambil keputusan layaknya manusia. Dalam proses pengembangan teknologi seperti ini, banyak cara yang diusahakan agar mendapatkan teknologi secerdas mungkin dengan tetap mempertahankan kecepatan dan keakuratan dari teknologi itu sendiri.

Salah satu metode untuk mengembangkan teknologi yang cerdas adalah dengan menggunakan jaringan saraf tiruan yang prinsip kerjanya diadaptasi dari cara kerja dari jaringan saraf biologi pada manusia, dimulai dari bagian penerima rangsang, proses berpikir, hingga proses mengambil keputusan. Untuk memperoleh jaringan saraf tiruan yang baik, maka jaringan ini perlu dilatih dengan sejumlah pola untuk dipelajari. Berbagai macam algoritma telah dikembangkan dalam usaha untuk mengoptimalkan dan memaksimalkan proses pembelajaran jaringan ini, baik dari segi keakuratan maupun dari segi kecepatan waktu komputasi. Namun, kecenderungan dari algoritma yang telah ada saat ini menunjukkan bahwa waktu pembelajaran bagi jaringan ini cukup lama walaupun hasil yang ditunjukkan sudah cukup baik.

Pada skripsi ini akan dibahas mengenai pengembangan algoritma *Radial Basis Function* pada jaringan saraf tiruan. Algoritma *Radial Basis Function* merupakan metode yang memanfaatkan jarak antara data dengan nilai tengah data dari hasil pengelompokan. Algoritma *Radial Basis Function* memiliki kelebihan yakni lebih sederhana serta lebih cepat waktu komputasinya jika dibandingkan dengan algoritma *Backpropagation*. Berdasarkan alasan tersebut, diharapkan dengan menggunakan algoritma *Radial Basis Function* pada jaringan saraf tiruan akan memiliki kelebihan kecepatan komputasi yang cepat.

1.2 Tujuan

Skripsi ini disusun berdasarkan riset dan percobaan yang dilakukan dalam usaha untuk mengembangkan algoritma *Radial Basis Function* yang diaplikasikan pada pembelajaran jaringan saraf tiruan.

Tujuan dari skripsi ini sendiri adalah sebagai berikut:

1. Mengembangkan algoritma *Radial Basis Function* untuk melatih jaringan saraf tiruan.
2. Mendapatkan kecepatan komputasi yang lebih baik dari jaringan saraf tiruan dengan algoritma *Backpropagation*.
3. Menganalisis proses pembelajaran jaringan saraf tiruan untuk algoritma *Radial Basis Function*.

1.3 Batasan Masalah

Fokus pembahasan dari penelitian ini adalah, kecepatan komputasi dan hasil pembelajaran jaringan saraf tiruan yang menggunakan algoritma *Radial Basis Function*. Untuk melakukan hal tersebut, dilakukan beberapa pengembangan perhitungan matematis dari koreksi nilai tengah (*center*) dan jarak antara data atau lebar data (*spread*). Setelah mendapatkan model matematis dari perhitungan yang telah dibuat, maka model ini diterapkan ke sebuah program jaringan saraf tiruan. Perangkat lunak yang digunakan untuk melakukan simulasi pada penelitian ini adalah MATLAB R2009a.

1.4 Metodologi Penulisan

Metodologi yang digunakan selama melakukan penelitian dan penulisan laporan skripsi ini adalah:

1. Studi Literatur

   Penulis membaca buku, jurnal, skripsi, serta literatur lain yang berkaitan dengan jaringan saraf tiruan algoritma *Radial Basis Function*.
2. Konsultasi dengan dosen pembimbing

   Penulis melakukan pertemuan untuk bimbingan dengan dosen pembimbing mengenai penelitian dan penulisan dari skripsi ini. Jika terdapat

permasalahan, maka dosen pembimbing akan mengarahkan dan memberikan solusi untuk membantu penulis.

3. Diskusi dengan orang yang lebih memahami mengenai peneilitian yang serupa

   Penulis bertanya dan mendiskusikan penelitian ini kepada pihak-pihak yang memahami penelitian yang dilakukan.

1.5 Sistematika Penulisan

Sistematika dari penulisan skripsi ini dibagi ke dalam 5 bab, antara lain sebagai berikut :

1. BAB 1 Pendahuluan

   Bab ini berisi Latar Belakang, Tujuan Penelitian, Batasan Masalah, Metodologi Penelitian, dan Sistematika Penulisan.

2. BAB 2 Dasar Teori Algoritma *Radial Basis Function*

   Pada bab ini, penulis menjelaskan mengenai dasar-dasar jaringan saraf tiruan, algoritma *Radial Basis Function*, serta data-data yang digunakan pada penelitian.

3. BAB 3 Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error* Kuadratis

   Bab ini akan membahas performa dari hal-hal yang berhubungan dengan jaringan saraf tiruan *Radial Basis Function* dengan fungsi e*rror* kuadratis sesuai dengan batasan-batasan percobaan.

4. BAB 4 Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy*

   Bab ini membahas performa dari hal-hal yang berhubungan dengan jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error cross-entropy* sesuai dengan batasan-batasan percobaan.

5. BAB 5 Komparasi Jaringan Saraf Tiruan Algoritma *Radial Basis Function* dengan Algoritma *Backpropagation*

   Bab ini berisi hasil komparasi dan analisis antara jaringan saraf tiruan *Radial Basis Function* fungsi e*rror* kuadratis dengan jaringan saraf tiruan *Backpropagation*, jaringan saraf tiruan *Radial Basis Function* fungsi *error*

cross-entropy dengan jaringan saraf tiruan *Backpropagation*, serta jaringan saraf tiruan *Radial Basis Function* fungsi e*rror* kuadratis dengan jaringan saraf tiruan *Radial Basis Function* fungsi *error cross-entropy*.

6. Kesimpulan

   Bab ini berisi kesimpulan dari penelitian yang berhubungan dengan tujuan dari penelitian ini sendiri.

BAB 2
PEMBELAJARAN JARINGAN SARAF TIRUAN ALGORITMA *RADIAL BASIS FUNCTION*

Pada bab ini akan dijelaskan mengenai pengertian jaringan saraf tiruan, jaringan saraf tiruan *Radial Basis Function*, *Z-score*, dan data yang digunakan untuk melakukan percobaan.

2.1 Pengertian Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan suatu kecerdasan buatan yang memproses informasi dengan cara meniru kinerja dari jaringan saraf biologis manusia. Metode kerjanya adalah dengan menerima rangsangan, mengolah rangsangan tersebut, dan mengambil keputusan berdasarkan dari pola-pola atau pengalaman yang telah dipelajari manusia.

Sedangkan menurut Hecht-Nielsen, definisi dari jaringan saraf tiruan (dengan beberapa penyesuaian) adalah sebagai berikut [2]:

> "*An Artificial Neural Network is a parallel, distributed information processing structure consisting of processing elements (which can process a local memory and carry out loaclized information processing operations) interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches ("fans out") into a many collateral connections as desired; each carries the same signal – the processing element output signal. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitarily with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via impinging connections and on values stored in the processing element's local memory.*"

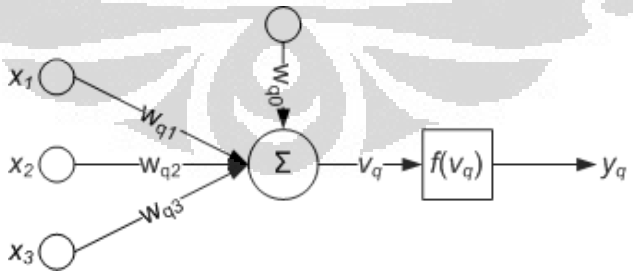
Setiap tipe jaringan saraf tiruan pasti memiliki minimal delapan elemen utama berikut [2]:

1. Sejumlah *processing elements* (PE) atau yang lebih dikenal dengan neuron.
2. *State of activation* untuk setiap neuron.
3. *Output function* untuk setiap neuron
4. Topologi jaringan (pola hubungan antar neuron).
5. *Propagation rule* atau *combining function* untuk menyebarkan aktivitas pada setiap neuron dalam jaringan.
6. *Activation rule* untuk meng-*update* aktivitas setiap neuron dengan suatu nilai pengaktif tertentu dan masukan dari neuron yang lain.
7. *External environment* yang menyediakan informasi untuk jaringan dan atau berinteraksi dengan jaringan.
8. *Learning rule* yang berguna untuk memodifikasi topologi jaringan berdasarkan informasi yang diperoleh dari *external environment*.

Secara garis besar hal-hal utama dari suatu jaringan saraf tiruan adalah neuron, topologi, dan pelatihan.

2.1.1 Neuron

Neuron merupakan bagian dalam jaringan saraf tiruan dengan seluruh proses perhitungan dilaksanakan dalam jaringan [2]. Bentuk umum dari sebuah neuron digambarkan dalam Gambar 2.1.



Gambar 2.1 Pengolahan Data pada Neuron

Jumlah masukan ke suatu neuron bervariasi untuk setiap neuron, bergantung pada koneksi antara neuron tersebut dengan neuron yang lain.

Sedangkan keluaran suatu neuron hanya satu. Dua hal penting yang harus dimiliki oleh sebuah neuron adalah [2]:

1. Neuron hanya membutuhkan informasi yang bersifat lokal. Semua informasi yang dibutuhkan oleh neuron untuk menghasilkan sebuah keluaran didapat dari masukan dan elemen lain dalam neuron itu sendiri tanpa terpengaruh oleh kondisi atau informasi lain dalam jaringan.
2. Neuron hanya menghasilkan sebuah keluaran yang akan dikirmkan ke neuron yang lain dalam jaringan atau sebagai keluaran dari jaringan.

Keluaran sebuah neuron merupakan fungsi dari masukan dan pembobotan terhadap setiap nilai masukan yang secara matematis dinyatakan dengan fungsi aktivasi.

Beberapa fungsi yang menyatakan hubungan keluaran dengan masukan pada suatu neuron, antara lain [5]:

1. Fungsi linear

$$\phi(x) = x \tag{2.1}$$

2. Fungsi pendekatan kubik

$$\phi(x) = x^3 \tag{2.2}$$

3. Fungsi *thin-plate-spline*

$$\phi(x) = x^2 \ln(x) \tag{2.3}$$

4. Fungsi invers multikuadratik

$$\phi(x) = \frac{1}{\sqrt{x^2 + \sigma^2}} \tag{2.4}$$

5. Fungsi multikuadratik

$$\phi(x) = \sqrt{x^2 + \sigma^2} \tag{2.5}$$

6. Fungi Gaussian

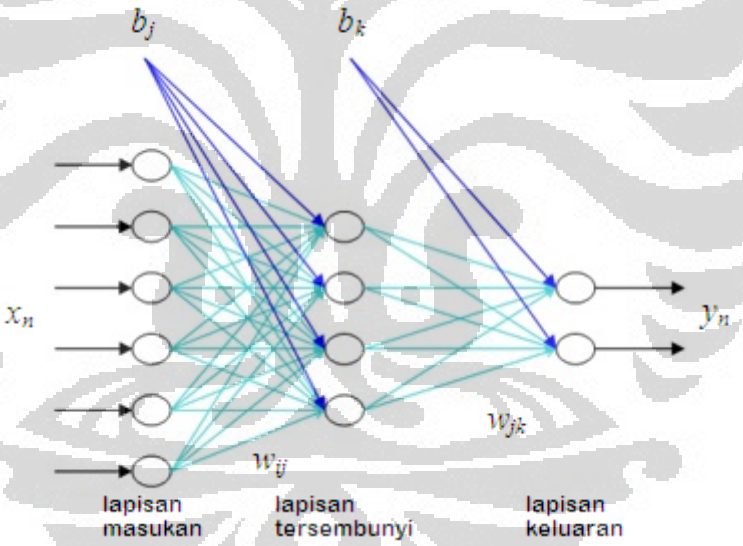
$$\phi(x) = e^{\left(\frac{-x^2}{2\sigma^2}\right)} \tag{2.6}$$

Dimana x adalah *mean* dan σ adalah variansi yang telah ditentukan sebelumnya.

2.1.2 Topologi Jaringan

Topologi jaringan saraf tiruan merupakan susunan neuron, koneksi, serta pola masukan dan keluaran yang disusun dalam beberapa lapisan agar setiap neuron dalam suatu lapisan memiliki dua hal yang menjadi karakteristik suatu lapisan yaitu [2]:

1. Koneksi masukan untuk setiap neuron dalam suatu lapisan berasal dari sumber yang sama, misalnya masukan untuk setiap neuron pada lapisan masukan berasal dari pola masukan.
2. Neuron dalam setiap lapisan menggunakan tipe *update dynamic* yang sama, misalnya setiap neuron memiliki pola koneksi dan fungsi transfer yang sama.



Gambar 2.2 Contoh Topologi Jaringan Saraf Tiruan

2.1.3 Pelatihan Jaringan Saraf Tiruan

Interkasi antara keluaran neuron dengan jaringan selama proses *learning* menghasilkan perubahan-perubahan dalam parameter-parameter jaringan (khususnya bobot) sesuai dengan pelatihannya. Secara umum jenis pelatihan dapat dibagi menjadi dua yakni pelatihan yang diarahkan (*supervised learning*) serta pelatihan yang tidak diarahkan (*unsupervised learning*) [2].

Pelatihan yang diarahkan merupakan pelatihan yang keluaran dari setiap neuron jaringan saraf tiruan diarahkan sesuai dengan kebutuhan. Pada pelatihan ini, untuk melakukan koreksi pada parameter-parameter yang mempengaruhi hasil dari jaringan dengan cara membandingkan keluaran yang diinginkan dengan keluaran dari jaringan serta selisih dari keduanya.

Sedangkan pelatihan yang tidak diarahkan merupakan pelatihan yang keluaran dari setiap neuron di jaringan saraf tiruan tidak dibandingkan dengan keluaran yang diinginkan, tetapi membandingkannya dengan keluaran yang mungkin dihasilkan. Pada bagian ini, jaringan menggunakan korelasi antarmasukan untuk mengubah-ubah parameter di dalamnya dengan tujuan untuk membentuk suatu kelompok masukan yang diharapkan dari sekelompok masukan yang mirip akan menghasilkan keluaran yang mirip pula.

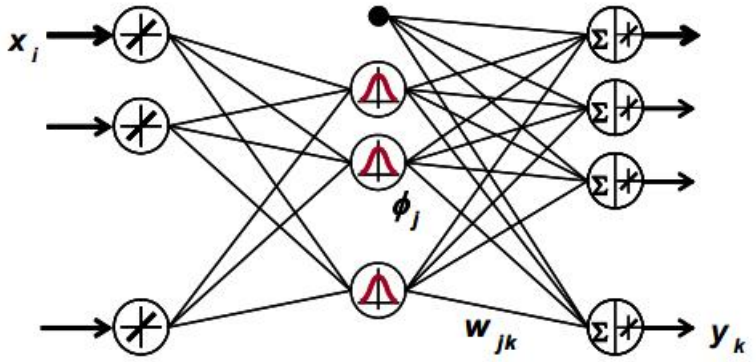
2.2 Pengertian Jaringan Saraf Tiruan *Radial Basis Function*

Model jaringan saraf tiruan *Radial Basis Function* merupakan salah satu bentuk *multilayer perceptron* yang memperbaiki nilai-nilai bobot, nilai tengah, dan jarak antar data agar mengurangi kesalahan yang terjadi pada keluaran jaringan. Pada model ini, Jaringan Saraf Tiruan menggunakan fungsi aktivasi basis (Gaussian) pada lapisan tersembunyi.

Dalam penerapannya untuk mendapatkan model jaringan saraf tiruan *Radial Basis Function* terbaik diperlukan kombinasi yang tepat antara jumlah variabel masukan, jumlah *node* (*cluster*) pada unit lapisan tersembunyi, nilai tengah serta standar deviasi (skala atau lebar data) dari variabel masukan pada setiap *node*, yang berimplikasi pada jumlah parameter yang optimal.

2.2.1 Arsitektur Jaringan Saraf Tiruan *Radial Basis Function*

Jaringan saraf tiruan *Radial Basis Function* tersusun dalam tiga lapisan (lapisan masukan, lapisan tersembunyi, dan lapisan keluaran). Jaringan saraf tiruan *Radial Basis Function* dapat digambarkan seperti pada Gambar 2.3.



Gambar 2.3 Arsitektur Jaringan Saraf Tiruan *Radial Basis Function*

Neuron-neuron lapisan tersembunyi pada jaringan saraf tiruan *Radial Basis Function* melakukan transformasi nonlinear dan memetakan masukan pada neuron masukan ke neuron tersembunyi tanpa parameter yang diubah-ubah. Selanjutnya neuron-neuron di lapisan keluaran melakukan kombinasi linear terhadap neuron tersembunyi dengan parameter yang diubah-ubah yakni bobot hubungan antara neuron di lapisan tersembunyi dengan neuron-neuron di lapisan keluaran. Nonlinearitas dalam jaringan saraf tiruan *Radial Basis Function* dapat dipilih dari beberapa fungsi nonlinear yang ada.

2.2.2 Fungsi *Error* pada Jaringan Saraf Tiruan *Radial Basis Function*

Telah dijelaskan pada bagian sebelumnya bahwa selisih antara keluaran jaringan dengan yang diinginkan akan digunakan untuk mengoreksi parameter-parameter yang berhubungan dengan pengambilan keputusan. Pada penelitian ini digunakan dua metode penghitungan kesalahan atau fungsi *error*. Fungsi *error* sendiri merupakan fungsi yang digunakan untuk memprediksi kesalahan yang terjadi antara keluaran jaringan dengan keluaran yang diinginkan. Biasanya fungsi *error* yang digunakan adalah fungsi yang sederhana, yakni fungsi *error* Lyapunov atau fungsi *error* kuadratis. Selain itu, pada penelitian ini juga akan digunakan fungsi e*rror cross-entropy* .

1. Fungsi *Error* Kuadratis

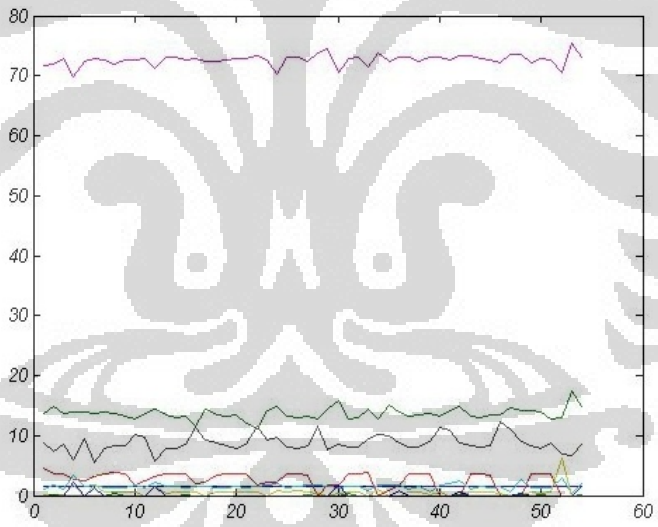
$$E = \frac{1}{L}\sum_{l=1}^{L}(T - Y)^2 \tag{2.7}$$

2. Fungsi *Error Cross-Entropy*

$$E = \frac{1}{L}\sum_{l=1}^{L}(-T\ln(Y) + (1-T)\ln(1-Y)) \tag{2.8}$$

2.3 *Z-Score*

Sebuah jaringan saraf tiruan dapat melakukan pemodelan baik untuk fungsi yang linear ataupun nonlinear. Fungsi-fungsi tersebut dapat dicari modelnya melalui proses pembelajaran menggunakan data-data masukan yang memiliki target. Namun, terdapat permasalahan dalam rentang data dan nilai rata-rata masing-masing dimensi. Tidak semua data memiliki rentang data dan nilai rata-rata yang sama. Kondisi ini akan membuat jaringan saraf lebih sulit untuk memodelkan suatu fungsi dengan data masukan tersebut.

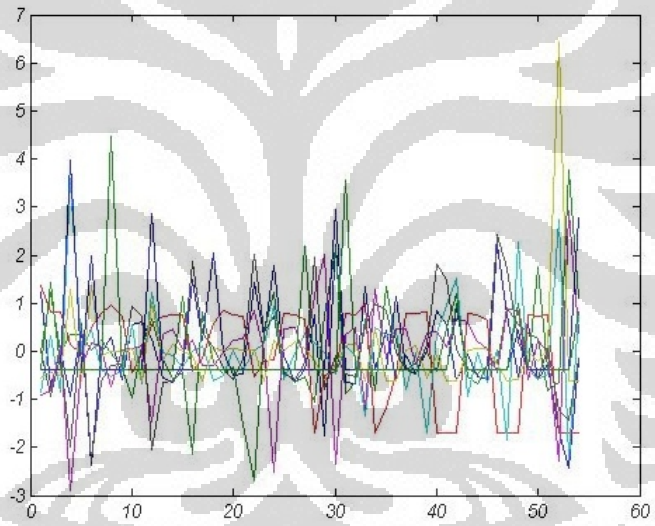


Gambar 2.4 Grafik Data Masukan dengan 7 Dimensi sebelum *Z-score*

Pada Gambar 2.4 terdapat contoh grafik data masukan dengan 7 dimensi. Data tersebut memiliki 7 buah garis yang saling terpisah, bahkan ada yang posisinya jauh terpisah dengan dimensi yang lain. Selain itu, lebar jangkauan masing-masing data juga berbeda. Garis yang berwarna ungu memiliki jangkauan yang paling lebar dan garis yang berwarna biru muda adalah data yang memiliki jangkauan paling kecil.

Jika Data pada Gambar 2.4 dimasukkan dan dimodelkan oleh jaringan saraf tiruan akan terdapat kesalahan atau *error* yang sangat besar saat kondisi pelatihan. Hal ini terjadi karena jaringan saraf tiruan akan lebih sulit untuk menentukan dimenasi mana yang lebih berpengaruh dan kurang berpengaruh pada keluaran jaringan. Oleh karena itu, untuk menyeragamkan nilai rata-rata dan jangkauan masing-masing dimensi, digunakan suatu metode yang disebut dengan *Z-score*.

Z-score dapat menghasilkan data yang lebih seragam dalam hal jangkauan dan rata-ratanya. Hasil keluaran dari *Z-score* dapat lebih mudah dikenali oleh jaringan saraf tiruan karena nilainya yang lebih seragam.



Gambar 2.5 Grafik Data Masukan dengan 7 Dimensi setelah *Z-score*

Metode *Z-score* menghitung nilai rata-rata dan standar deviasi dari masing-masing dimensi data awal. Kemudian, data tersebut akan dikurangi dengan rata-ratanya agar nilai rata-rata seluruh dimensi menjadi 0. Setelah dikurangi, data masih memiliki nilai jangkauan yang berbeda-beda. Oleh karena itu, data kemudian dibagi dengan standar deviasi agar jangkauan data menjadi lebih seragam. Secara umum, persamaan untuk fungsi *Z-score* adalah:

$$Z = \frac{x - \dot{x}}{\sigma} \tag{2.9}$$

2.4 Data Percobaan

Pada skripsi ini, percobaan dilakukan dengan menggunakan 5 macam set data, yaitu 4 macam set data yang didapat dari "*UCI Repository of Machine Learning Database*" berupa set data *heart*, *ionosphere*, *iris*, dan *sonar* serta 1 set data uranium dari BATAN yang diperoleh dari saudara Dede Sutarya. Data tersebut dipilih karena memiliki jumlah data serta dimensi yang berbeda-beda agar dapat diketahui pengaruhnya terhadap Jaringan Saraf Tiruan *Radial Basis Function*.

Berikut ini akan diuraikan mengenai set data yang digunakan:

1. *Heart Diseases Database*

   Informasi : Data ini diperoleh dari sebuah eksperimen yang bertujuan untuk menganalisis seorang pasien menderita penyakit jantung atau tidak.

   Jumlah data : 270

   Jumlah kelas : 2 (kelas 1 = 150 dan kelas 2 = 170)

   Jumlah atribut : 14

   Informasi atribut : 1. Umur
   2. Jenis kelamin
   3. Jenis sakit di dada
   4. Tekanan darah saat beristirahat
   5. Serum *cholesterol* dalam mg/dl
   6. Gula darah saat istirahat > 120 mg/dl
   7. Hasil *electrocardiographic* saat istirahat (0 - 2)
   8. Kecepatan detak jantung maksimum
   9. *Exercise induced angina*
   10. *Oldpeak* = depresi ST yang diakibatkan oleh olah raga relative terhadap saat istirahat
   11. Kemiringan dari puncak latihan segmen ST
   12. Banyaknya nadi utama (0 – 3) yang diwarnai oleh *flourosopy*
   13. Kondisi: 3 = normal, 6 = cacat tetap, 7: cacat sementara

14. Tipe kelas: 1 untuk tidak menderita penyakit jantung dan 2 untuk menderita penyakit jantung.

2. *Ionosphere Database*

Penulis : Jhon Hopkins University (1989)

Informasi : Data ini diperoleh dari sistem di Goose Bay, Labrador. Sistem ini terdiri dari 16 antena frekuensi tinggi. Target dari sistem ini adalah elektron yang ada di lapisan ionosfer. Apabila dinyatakan dengan kata "*good*" menandakan adanya struktur tertentu di ionosfer sedangkan "*bad*" menandakan tidak adanya struktur di sana.

Jumlah data : 351

Jumlah kelas : 2 (kelas 1 = 225dan kelas 2 = 126)

Jumlah atribut : 35

Informasi atribut : Atribut 1 sampai dengan 34 merupakan informasi hasil pengolahan data dan atribut ke-35 merupakan kelas yang berisi "*good*" dan "*bad*".

3. *Iris Plants Database*

Penulis : R. A. Fisher (Juli 1988)

Informasi : Data ini terdiri dari 3 kelas dengan 50 data untuk masing-masing kelas yang setiap kelasnya mewakili sebuah jenis bunga iris yakni Iris Setosa, Iris Versicolor, dan Iris Virginica. Setiap kelas dapat dipisahkan secara linear dari yang lainnya.

Jumlah data : 150 (50 Iris Setosa, 50 Iris Versicolor, dan 50 Iris Virginia)

Jumlah kelas : 3 (masing-masing kelas = 50)

Jumlah atribut : 5

Informasi atribu : 1. Panjang mahkota bunga dalam ukuran centimeter

2. Tebal mahkota bunga dalam ukuran centimeter

3. Panjang kelopak bunga dalam ukuran centimeter
4. Tebal kelopak bunga dalam ukuran centimeter
5. Jenis kelas: 1 untuk Iris Sentosa, 2 untuk Iris Versicolour, dan 3 untuk Iris Virginica

4. *Sonar Database*

Informasi : Data ini merupakan rekaman dari tingkat energi pada 60 frekuensi yang berbeda untuk menentukan perbedaan antara benda metal dan batu.
Jumlah data : 208 (111 benda metal dan 97 benda batu)
Jumlah kelas : 2 (kelas 1 = 111 dan kelas 2 = 107)
Jumlah atribut : 61
Informasi atribut : Atribut 1 sampai dengan 60 merupakan tingkat energi pada 60 frekuensi yang berbeda dan atribut ke-61 menunjukkan jenis kelas (M untuk metal dan R untuk batu).

5. Uranium Database

Penulis : Dede Sutarya (BATAN)
Informasi : Data ini merupakan penggolongan kualitas uranium dalam pengendalian fabrikasi bahan bakar nuklir.
Jumlah data : 150
Jumlah kelas : 3 (masing-masing kelas = 50)
Jumlah atribut : 6
Informasi atribut : 1. Tinggi (dalam centimeter)
2. Volume (dalam cc)
3. Berat (dalam gram)
4. Kepadatan butiran (gram/cc)
5. Kepadatan butiran secara teori
6. Kelas 1: uranium klasifikasi 1, Kelas 2: uranium klasifikasi 2, dan Kelas 3: uranium klasifikasi 3

BAB 3
JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN FUNGSI *ERROR* KUADRATIS

Pada bagian ini akan dijelaskan mengenai tujuan, algoritma, grafik hasil percobaan, serta analisis dari jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis.

3.1 Tujuan Percobaan JST RBF dengan Fungsi *Error* Kuadratis

Tujuan yang ingin dicapai pada percobaan yang dilakukan adalah sebagai berikut:

1. Memahami kinerja dari jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis.
2. Mengetahui akibat-akibat dari perubahan parameter-parameter yanga ada pada jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis.

3.2 Prosedur Percobaan JST RBF dengan Fungsi *Error* Kuadratis

Pada percobaan ini, yang pertama kali dilakukan adalah mengumpulkan data serta menyusunnya. Data-data tersebut harus telah mengalami proses *z-score* agar lebih mempermudah jaringan dalam mengolahnya. Selanjutnya, data-data tersebut dilatih pada jaringan dengan rasio data 50% dari total tiap set data secara keseluruhan. Jaringan akan berhenti melakukan pelatihan jika telah mencapai batas *error* minimum sebesar 0.01 atau epoh maksimal sebesar 10000. Setelah mengalami pelatihan, maka dilakukan pengujian pada data yang digunakan untuk pelatihan dan 50% set data yang tidak digunakan dalam pelatihan. Diagram alur algoritma jaringan dapat dilihat pada subbab 3.3. Untuk mendapatkan data dari hasil percobaan, digunakan komputer dengan spesifikasi sebagai berikut:

Prosesor	: Intel Xeon T7100 @2.66 GHz
Memori	: 3328 MB RAM
Sistem Operasi	: Windows XP Professional SP 3
Perangkat Lunak	: MATLABR2009a

Pada jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis, terjadi 2 kali proses propagasi yaitu propagasi maju dan propagasi balik. Untuk propagasi maju, proses yang terjadi adalah data masukan masuk ke jaringan melalui lapisan masukan dan langsung menuju ke lapisan tersembunyi. Di lapisan tersembunyi, data masukan dari setiap neuron di lapisan masukan mengalami perhitungan antara data masukan dengan nilai tengah data masukan dan kemudian dengan fungsi aktivasi Gaussian.

$$Z_{in} = \frac{\sum (x-c)^2}{\sigma^2} \tag{3.1}$$

$$Z = e^{-Z_{in}} \tag{3.2}$$

Keluaran dari lapisan tersembunyi ini mengalami perhitungan dengan pengaruh dari bias serta bobot antara lapisan tersembunyi dengan lapisan keluaran. Lalu keluaran dari lapisan keluaran adalah hasil dari perhitungan fungsi sigmoid unipolar.

$$Y_{in} = w_{0l} + \sum_{m=1}^{M} Z \cdot w_{ml} \tag{3.3}$$

$$Y = \frac{1}{(1+e^{-Y_{in}})} \tag{3.4}$$

Pada jaringan saraf tiruan *Radial Basis Function* ini, setelah propagasi maju selesai, dihitung besarnya kesalahan yang terjadi pada keluaran terhadap target yang bersesuaian.

$$E = \frac{1}{2}\sum_{l=1}^{L} (T-Y)^2 \tag{3.5}$$

Saat propagasi balik, terdapat koreksi nilai-nilai dari bobot dan bias dari lapisan tersembunyi ke lapisan keluaran serta nilai tengah dan lebar data. Koreksi dari nilai-nilai tersebut didapat dari turunan rumus *error* terhadap komponen yang

ingin dikoreksi. Berikut ini akan diperlihatkan turunan-turunan dari setiap fungsi yang akan digunakan pada proses propagasi balik.

1. Turunan fungsi *error* terhadap keluaran dari lapisan keluaran:

$$\frac{\partial E}{\partial Y}\frac{\partial Y}{\partial Y_{in}} = -(T-Y)Y(1-Y) \tag{3.6}$$

2. Turunan fungsi masukan dari lapisan keluaran terhadap fungsi keluaran pada lapisan tersembunyi:

$$\frac{\partial Y_{in}}{\partial Z} = \sum_{m=1}^{M} w_{ml} \tag{3.7}$$

3. Turunan fungsi keluaran dari lapisan tersembunyi terhadap fungsi masukan pada lapisan tersembunyi dan persamaan (3.2) disubtitusi ke dalam hasil turunan, maka didapat:

$$\frac{\partial Z}{\partial Z_{in}} = -e^{-Z_{in}} = -Z \tag{3.8}$$

4. Turunan fungsi masukan dari lapisan tersembunyi terhadap nilai tengah:

$$\frac{\partial Z_{in}}{\partial c} = \frac{-\sum(x-c)}{\sigma^2} \tag{3.9}$$

5. Turunan fungsi masukan dari lapisan tersembunyi terhadap nilai lebar data:

$$\frac{\partial Z_{in}}{\partial \sigma} = \frac{\sum(x-c)^2}{\sigma^3} \tag{3.10}$$

6. Turunan fungsi masukan dari lapisan tersembunyi terhadap bobot antara lapisan tersembunyi dengan lapisan kelauran:

$$\frac{\partial Y_{in}}{\partial w_{ml}} = Z \tag{3.11}$$

Berdasarkan dari turunan-turunan rumus sebelumnya, maka untuk koreksi nilai bobot akan didapat turunan rumus terhadap *error* sebagai berikut:

$$\frac{\partial E}{\partial w_{ml}} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial w_{ml}}$$

$$\frac{\partial E}{\partial w_{ml}} = -(T - Y) \times Y(1 - Y) \times Z \tag{3.12}$$

Untuk koreksi nilai tengah, maka turunan rumus terhadap *error* sebagai berikut:

$$\frac{\partial E}{\partial c} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial Z} \times \frac{\partial Z}{\partial Z_{in}} \times \frac{\partial Z_{in}}{\partial c}$$

$$\frac{\partial E}{\partial c} = -(T - Y) \times Y(1 - Y) \times \sum_{m=1}^{M} w_{ml} \times -Z \times \frac{\sum (x - c)}{\sigma^2} \tag{3.13}$$

Untuk koreksi nilai lebar data, maka turunan rumusnya sebagai berikut:

$$\frac{\partial E}{\partial \sigma} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial Z} \times \frac{\partial Z}{\partial Z_{in}} \times \frac{\partial Z_{in}}{\partial \sigma}$$

$$\frac{\partial E}{\partial \sigma} = -(T - Y) \times Y(1 - Y) \times \sum_{m=1}^{M} w_{ml} \times -Z \times \frac{\sum (x - c)^2}{\sigma^3} \tag{3.14}$$

3.3 Algoritma Percobaan JST RBF dengan Fungsi *Error* Kuadratis

Algoritma yang digunakan dalam melakukan pemrograman jaringan saraf tiruan dengan algoritma *Radial Basis Function* dengan fungsi e*rror* kuadratis adalah sebagai berikut:

1. Inisialisasi
    a) Inisialisasi bobot dan bias untuk lapisan tersembunyi ke lapisan keluaran dengan menggunakan metode Nguyen-Widrow.
    b) Menentukan vektor masukan dan vektor target, vektor nilai tengah RBF serta lebar data RBF.

2. Apabila kondisi berhenti belum terpenuhi, jalankan langkah 3-5.

3. Melakukan proses propagasi maju
   a) Setiap unit *input* menerima sinyal masukan dan menyebarkan sinyal tersebut ke lapisan tersembunyi.
   b) Setiap unit tersembunyi mengalami perhitungan terhadap vektor nilai tengah. Nilai Z_{in} dapat dicari dengan:

$$Z_{in} = \frac{\sum (x-c)^2}{\sigma^2} \tag{3.15}$$

   c) Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit tersembunyi. Nilai Z dapat dicari dengan:

$$Z = e^{-Z_{in}} \tag{3.16}$$

   d) Setiap unit keluaran menjumlahkan sinyal masukan yang telah diberi bobot. Nilai Y_{in} dapat dicari menggunakan persamaan:

$$Y_{in} = w_{0l} + \sum_{m=1}^{M} Z \cdot w_{ml} \tag{3.17}$$

   e) Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran dari unit keluaran. Nilai Y didapatkan dari persamaan berikut:

$$Y = \frac{1}{(1+e^{-Y_{in}})} \tag{3.18}$$

4. Melakukan propagasi balik
   a) Setiap unit keluaran menerima sebuah pola target yang bersesuaian dengan pola pelatihan masukan lalu menghitung informasi kesalahan

$$\delta = -(T-Y)(Y(1-Y)) \tag{3.19}$$

b) Menghitung koreksi bobotnya dan menghitung koreksi biasnya dengan persamaan:

$$\Delta w_{ml} = \alpha \cdot \delta \cdot Z \tag{3.20}$$

$$\Delta w_{0l} = \alpha \cdot \delta \tag{3.21}$$

c) Menghitung koreksi nilai tengah dengan persamaan:

$$\Delta c = \beta \cdot \delta \cdot \left(\sum_{m=1}^{M} w_{ml} \right) (-Z) \frac{-\sum (x - c)}{\sigma^2} \tag{3.22}$$

Untuk menghitung koreksi nilai lebar data RBF, dilakukan perhitungan awal pada setiap kelas pada set data masukan (1 sampai dengan n) dengan persamaan:

$$\Delta \sigma_n = \gamma \cdot \delta \cdot \left(\sum_{m=1}^{M} w_{ml} \right) (-Z) \frac{\sum (x - c)^2}{\sigma^3} \tag{3.23}$$

Setelah itu, maka dihitung nilai rata-rata dari Δσ menggunakan persamaan:

$$\Delta \sigma = \Delta \sigma_1 + \Delta \sigma_2 + ... + \Delta \sigma_n \tag{3.24}$$

5. Pengubahan nilai-nilai bobot, bias, nilai tengah, dan lebar RBF dengan persamaan:

$$w_{ml} = w_{ml} + \Delta w_{ml} \tag{3.25}$$

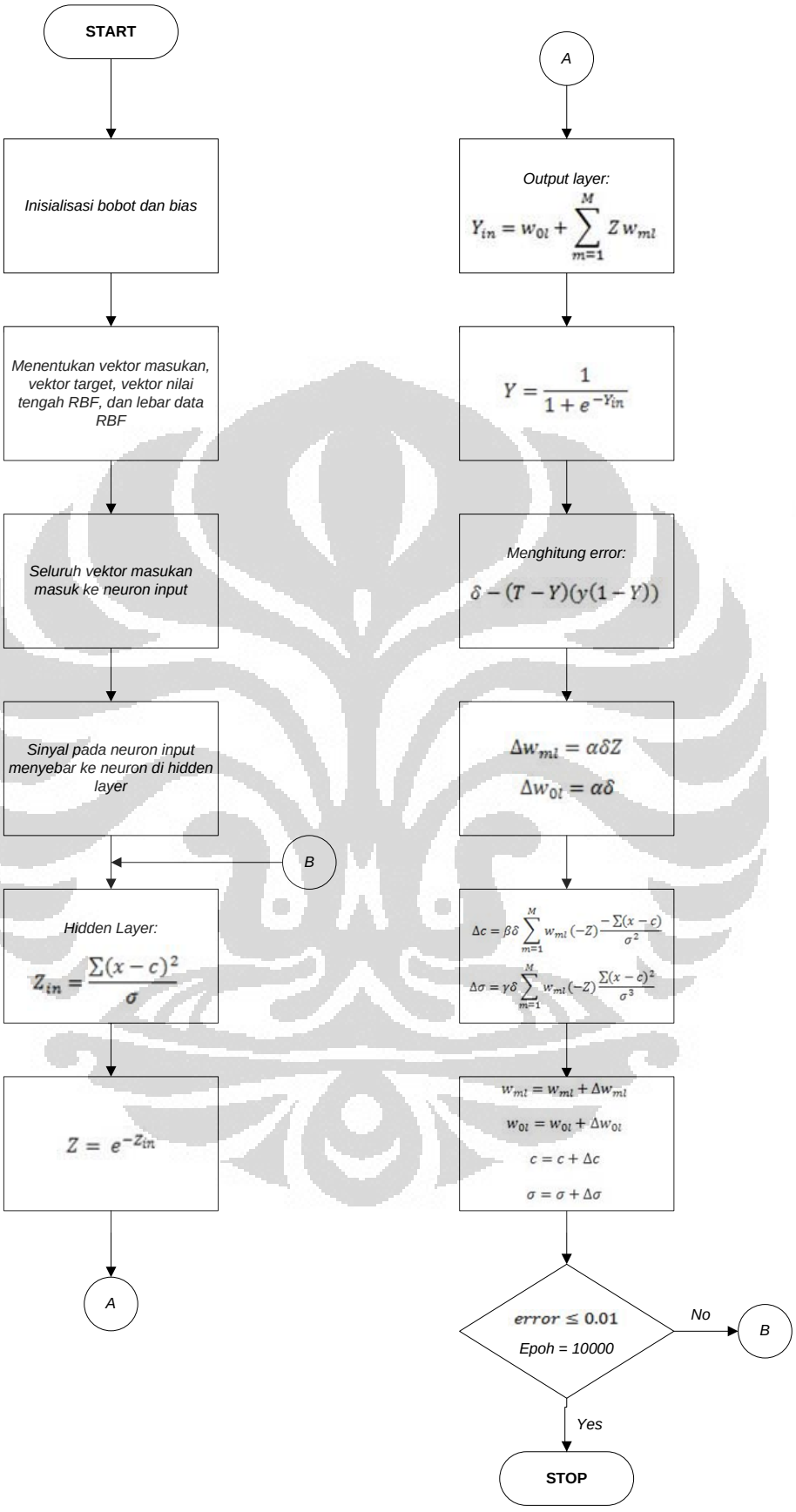
$$w_{0l} = w_{0l} + \Delta w_{0l} \tag{3.26}$$

$$c = c + \Delta c \tag{3.27}$$

$$\sigma = \sigma + \Delta \sigma \tag{3.28}$$

6. Pengujian kondisi berhenti.

   Berhenti jika *error* telah mencapai 0.01 atau epoh telah mencapai 10000.

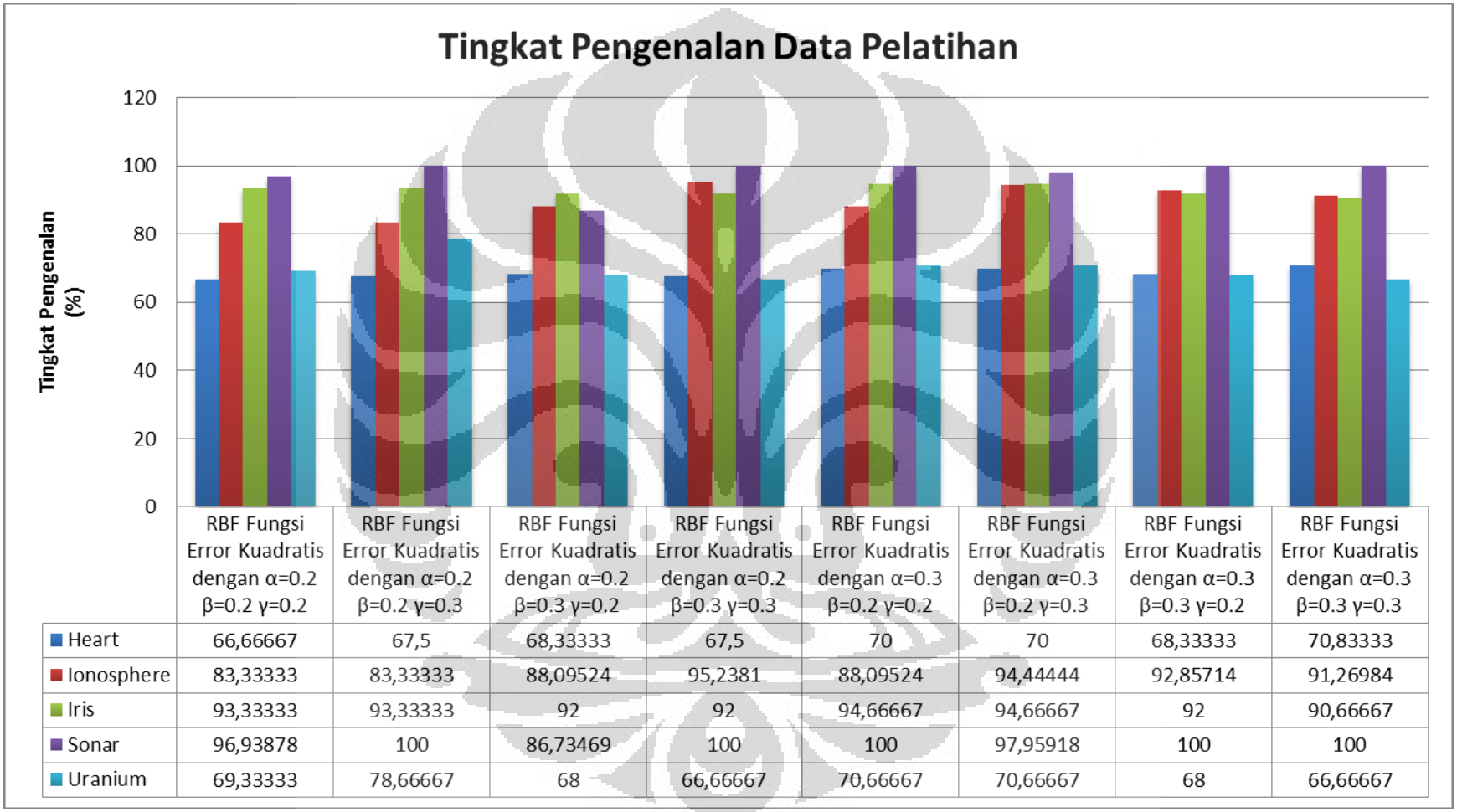


Gambar 3.1 Diagram Blok Algoritma JST RBF Fungsi *Error* Kuadratis

3.4 Hasil Percobaan JST RBF dengan Fungsi *Error* Kuadratis

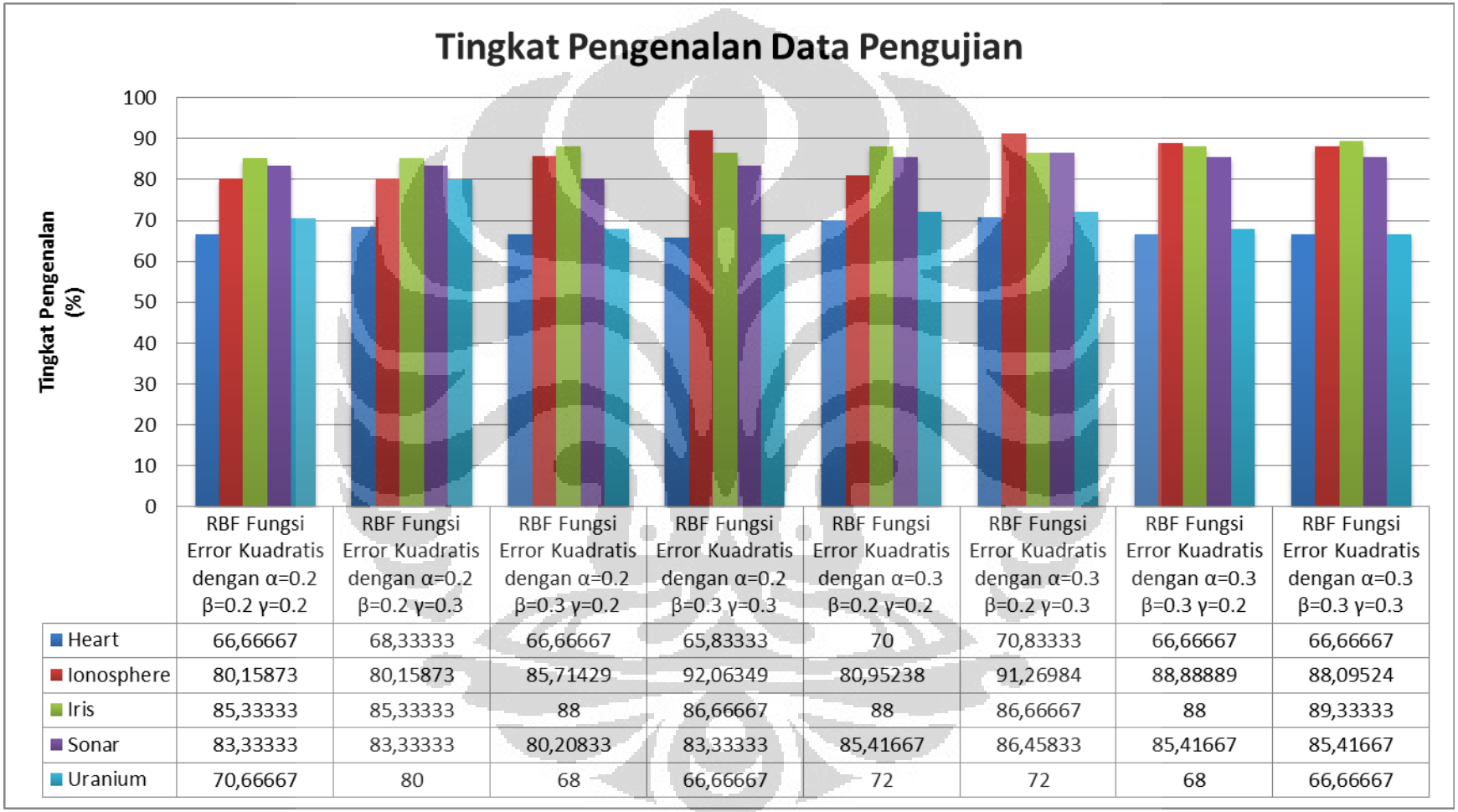
Pada bagian ini akan diperlihatkan grafik-grafik dari hasil pengolahan data yang telah didapat dari jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis. Parameter-parameter yang digunakan untuk mendapatkan hasil percobaan tersebut adalah sebagai berikut:

Rasio data untuk pelatihan (dalam %)	= 50
Variasi konstanta pembelajaran (α)	= 0.2 dan 0.3
Variasi konstanta pembelajaran nilai tengah (β)	= 0.2 dan 0.3
Variasi konstanta pembelajaran lebar data (γ)	= 0.2 dan 0.3
Kondisi stop *error* minimum	= 0.01
Kondisi stop epoh maksimum	= 10000



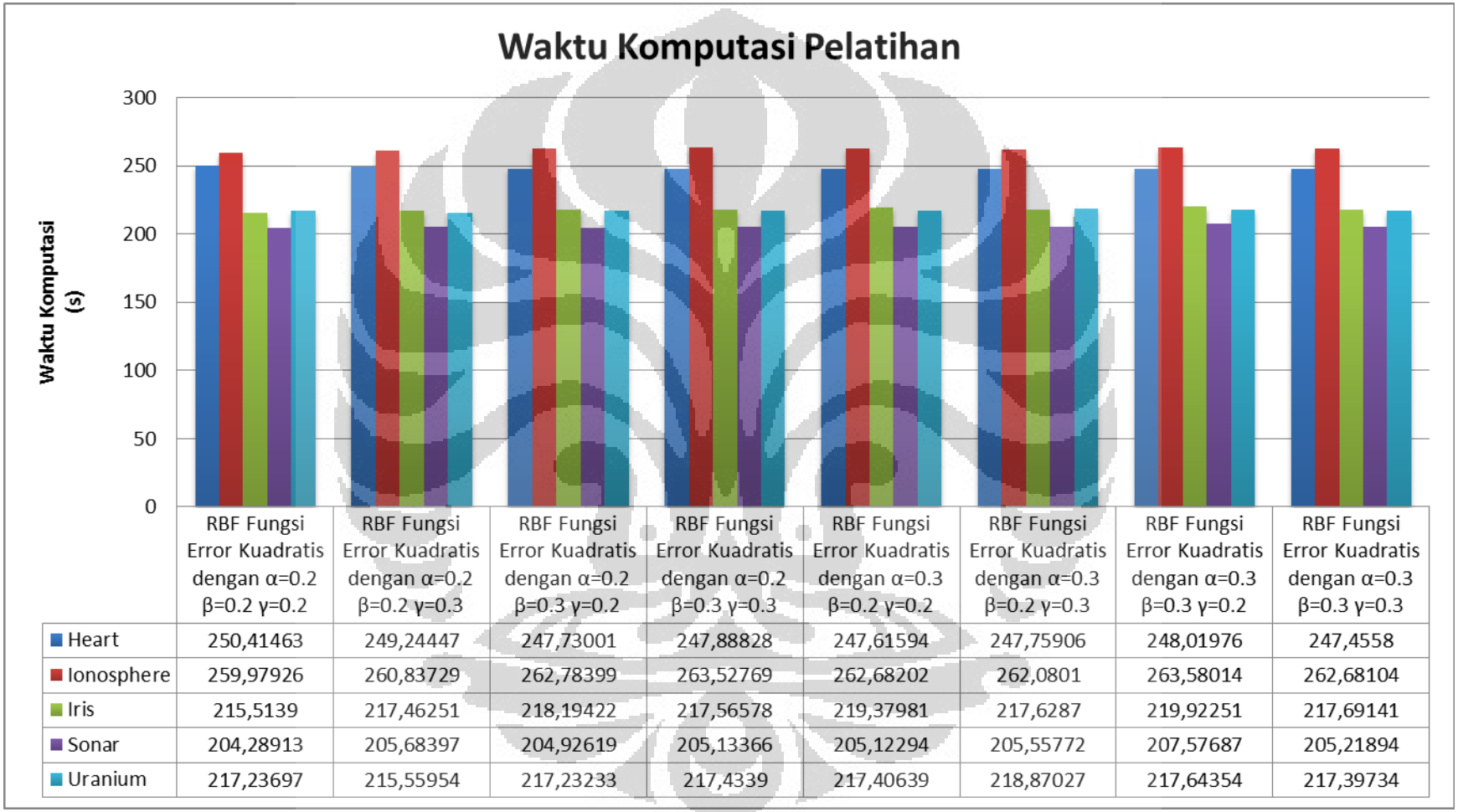
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	66,66667	67,5	68,33333	67,5	70	70	68,33333	70,83333
Ionosphere	83,33333	83,33333	88,09524	95,2381	88,09524	94,44444	92,85714	91,26984
Iris	93,33333	93,33333	92	92	94,66667	94,66667	92	90,66667
Sonar	96,93878	100	86,73469	100	100	97,95918	100	100
Uranium	69,33333	78,66667	68	66,66667	70,66667	70,66667	68	66,66667

Gambar 3.2 Tingkat Pengenalan Data Pelatihan pada JST RBF Fungsi *Error* Kuadratis



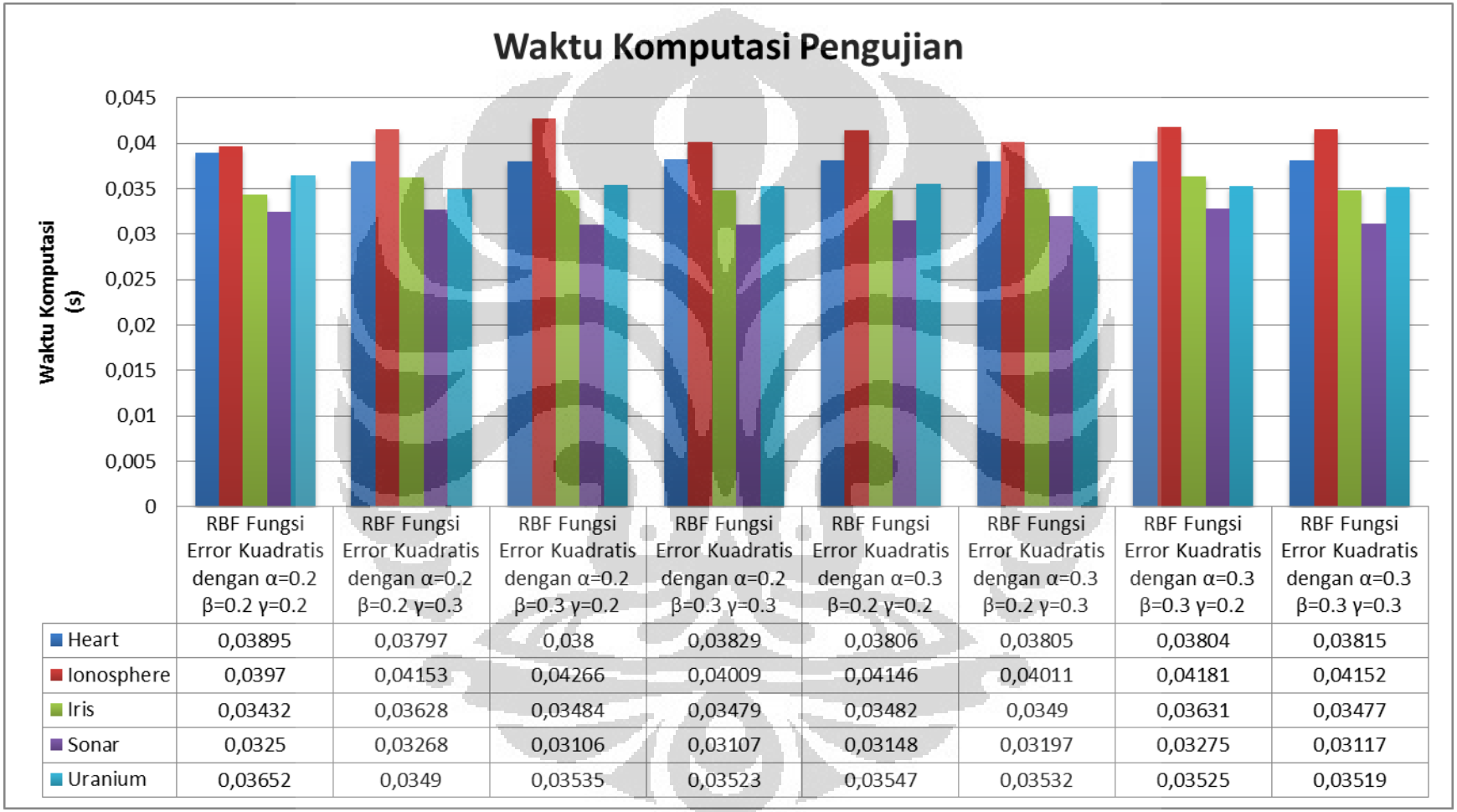
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	66,66667	68,33333	66,66667	65,83333	70	70,83333	66,66667	66,66667
Ionosphere	80,15873	80,15873	85,71429	92,06349	80,95238	91,26984	88,88889	88,09524
Iris	85,33333	85,33333	88	86,66667	88	86,66667	88	89,33333
Sonar	83,33333	83,33333	80,20833	83,33333	85,41667	86,45833	85,41667	85,41667
Uranium	70,66667	80	68	66,66667	72	72	68	66,66667

Gambar 3.3 Tingkat Pengenalan Data Pengujian pada JST RBF Fungsi *Error* Kuadratis



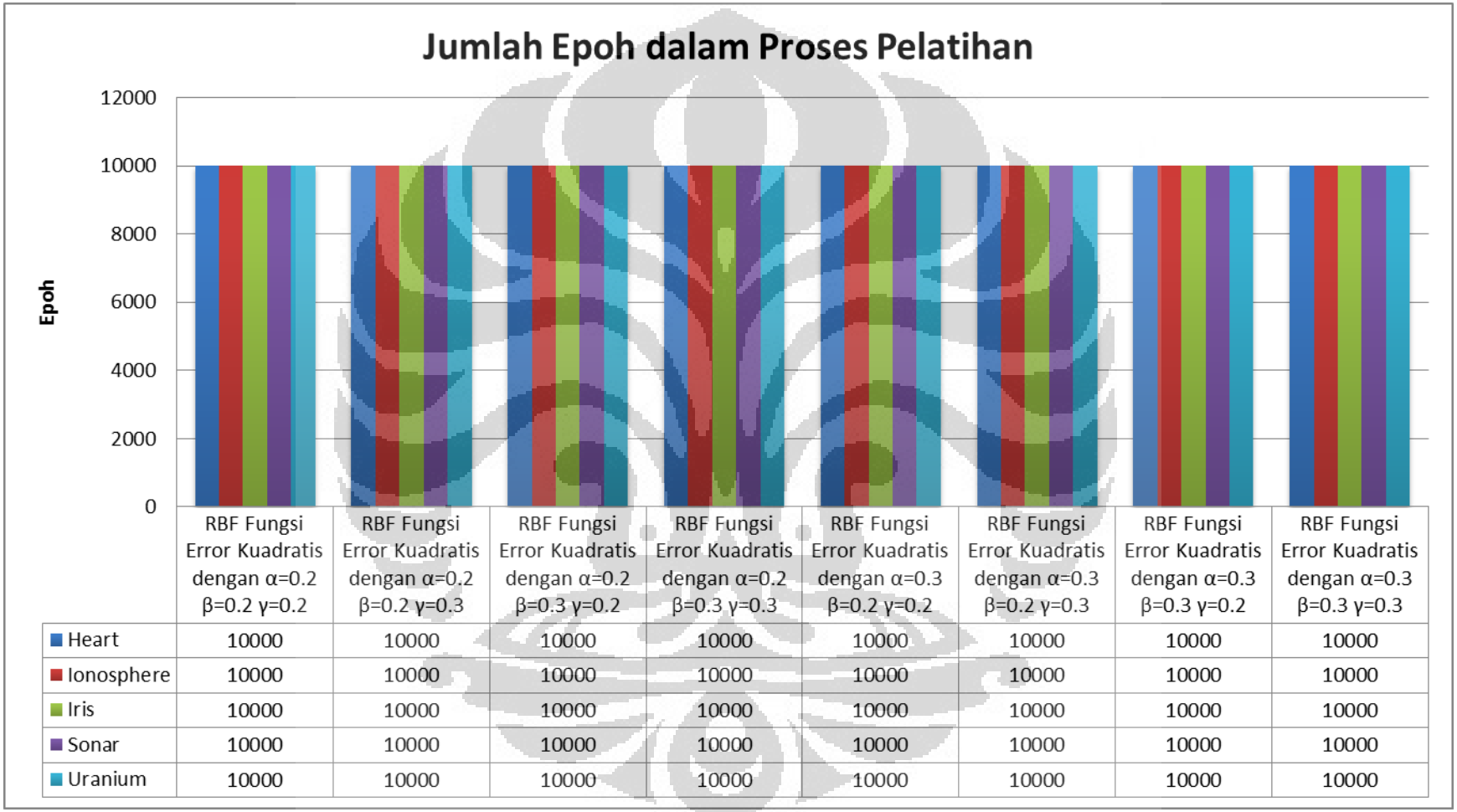
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	250,41463	249,24447	247,73001	247,88828	247,61594	247,75906	248,01976	247,4558
Ionosphere	259,97926	260,83729	262,78399	263,52769	262,68202	262,0801	263,58014	262,68104
Iris	215,5139	217,46251	218,19422	217,56578	219,37981	217,6287	219,92251	217,69141
Sonar	204,28913	205,68397	204,92619	205,13366	205,12294	205,55772	207,57687	205,21894
Uranium	217,23697	215,55954	217,23233	217,4339	217,40639	218,87027	217,64354	217,39734

Gambar 3.4 Waktu Komputasi Pelatihan pada JST RBF Fungsi *Error* Kuadratis



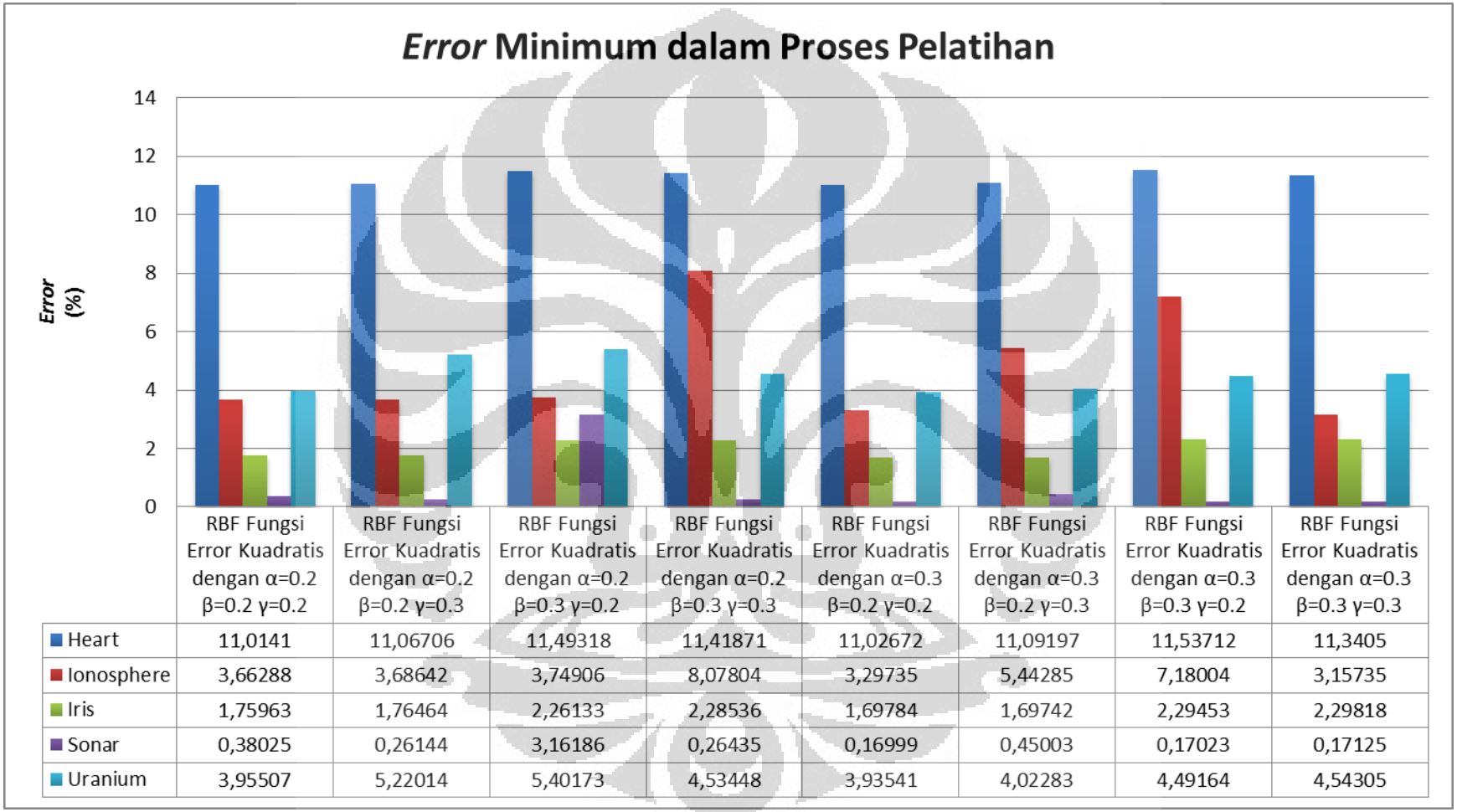
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	0,03895	0,03797	0,038	0,03829	0,03806	0,03805	0,03804	0,03815
Ionosphere	0,0397	0,04153	0,04266	0,04009	0,04146	0,04011	0,04181	0,04152
Iris	0,03432	0,03628	0,03484	0,03479	0,03482	0,0349	0,03631	0,03477
Sonar	0,0325	0,03268	0,03106	0,03107	0,03148	0,03197	0,03275	0,03117
Uranium	0,03652	0,0349	0,03535	0,03523	0,03547	0,03532	0,03525	0,03519

Gambar 3.5 Waktu Komputasi Pengujian pada JST RBF Fungsi *Error* Kuadratis



	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	10000	10000	10000	10000	10000	10000	10000	10000
Ionosphere	10000	10000	10000	10000	10000	10000	10000	10000
Iris	10000	10000	10000	10000	10000	10000	10000	10000
Sonar	10000	10000	10000	10000	10000	10000	10000	10000
Uranium	10000	10000	10000	10000	10000	10000	10000	10000

Gambar 3.6 Jumlah Epoh dalam Proses Pelatihan pada JST RBF Fungsi *Error* Kuadratis



	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3
Heart	11,0141	11,06706	11,49318	11,41871	11,02672	11,09197	11,53712	11,3405
Ionosphere	3,66288	3,68642	3,74906	8,07804	3,29735	5,44285	7,18004	3,15735
Iris	1,75963	1,76464	2,26133	2,28536	1,69784	1,69742	2,29453	2,29818
Sonar	0,38025	0,26144	3,16186	0,26435	0,16999	0,45003	0,17023	0,17125
Uranium	3,95507	5,22014	5,40173	4,53448	3,93541	4,02283	4,49164	4,54305

Gambar 3.7 *Error* Minimum dalam Proses Pelatihan pada JST RBF Fungsi *Error* Kuadratis

3.5 Analisis Percobaan JST RBF dengan Fungsi *Error* Kuadratis

Pada percobaan kali ini akan diamati hasil dari jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error* kuadratis. Data yang terdapat pada bagian tingkat pengenalan data pelatihan dan pengujian, waktu komputasi pelatihan dan pengujian, jumlah epoh dalam proses pelatihan, serta *error* minimum yang dicapai dalam proses pelatihan. Pada tingkat pengenalan data pelatihan (Gambar 3.2), didapat hasil yang baik untuk set data *ionosphere*, *iris*, dan *sonar* yaitu dengan nilai rata-rata tiap set data yang berkisar pada 90% bahkan untuk data *sonar* ada yang mencapai 100%. Sedangkan untuk set data *heart* dan uranium, hasil tingkat pengenalannya bernilai rata-rata 70% dengan tingkat pengenalan terbaik untuk untuk set data uranium 78.67% dan untuk set data *heart* sebesar 70.83%. Pada tingkat pengenalan data pengujian (Gambar 3.3) terjadi hal yang sama seperti pada tingkat pengenalan data pelatihan tetapi besarnya tingkat pengenalan untuk 3 set data pada *heart*, *ionosphere*, dan *sonar* lebih rendah dari pada tingkat pengenalan data pelatihan yakni bernilai rata-rata berkisar 85%. Set data *heart* dan uranium memiliki nilai rata-rata tingkat pengenalan data pengujian pada seluruh percobaan berkisar di 70%.

Waktu komputasi terbagi 2, yaitu waktu komputasi pelatihan (Gambar 3.4) dan waktu komputasi pengujian (Gambar 3.5). Waktu komputasi pelatihan merupakan waktu yang digunakan untuk melakukan proses komputasi pelatihan pada jaringan. Waktu komputasi pengujian merupakan waktu yang digunakan untuk melakukan proses komputasi pengujian pada jaringan. Keduanya memiliki waktu yang berbeda karena pada waktu komputasi pelatihan proses yang terjadi pada jaringan adalah proses propagasi maju dan propagsi balik sehingga memakan waktu yang lebih lama dibandingkan dengan waktu komputasi pengujian yang hanya melakukan proses komputasi maju dan menyeleksi data yang sesuai dengan target. Pada set data *sonar* memiliki waktu komputasi pelatihan yang tercepat di antara kelima set data dengan waktu komputasi rata-rata sebesar 205 detik. Set data yang memiliki waktu komputasi pelatihan terlama adalah set data *ionosphere* karena memiliki 252 data untuk tiap dimensi. Waktu komputasi pengujian yang tercepat terdapat pada set data *sonar* dengan besar nilai rata-rata 0.032. Waktu komputasi terlama terdapat pada set data *ionosphere* dengan waktu komputasi

rata-rata sebesar 0.041. Perbedaan waktu komputasi berbeda-beda untuk setiap set data karena terdapat faktor-faktor yang mempengaruhi seperti jumlah kelas serta jumlah data yang diolah dari tiap set data, keduanya akan mempengaruhi jumlah neuron pada lapisan tersembunyi. Neuron-neuron pada lapisan tersembunyi memiliki fungsi untuk menghitung jarak antara satu data terhadap nilai tengah dan lebar data untuk setiap kelas, sehingga menyebabkan banyak terjadinya perhitungan aritmatika yang mempengaruhi lamanya komputasi. Selain itu, jumlah epoh (Gambar 3.6) dan batas minimum *error* pun mempengaruhi waktu komputasi karena jumlah epoh berguna sebagai batas dari jumlah pengulangan proses pelatihan yang terjadi pada percobaan dan batas minimum *error* juga berguna sebagai batas untuk kondisi berhenti dari jaringan jika telah mencapai *error* yang diinginkan. Jika digunakan jumlah epoh yang berbeda-beda untuk melakukan percobaan, maka akan diperoleh waktu komputasi yang berbeda-beda pula. Pada percobaan ini, jumlah epoh yang digunakan sebagai batas maksimum pengulangan proses pelatihan sebesar 10000 epoh, sehingga jaringan akan berhenti jika telah mencapai batas epoh tersebut.

Selain hal-hal di atas yang telah disebutkan, pada percobaan didapatkan juga *error* minimum (Gambar 3.7). *Error* minimum yang terkecil terdapat pada set data *sonar* dengan nilai rata-rata 0.38% sedangkan *error* minimum yang terbesar terdapat pada set data *heart* dengan nilai rata-rata 11%. Pada pencapaian *error* minimum, fungsi *error* yang digunakan pada proses pelatihan sangat berpengaruh karena perhitungan dari fungsi *error* akan menghasilkan *error* antara data dengan target yang diharapkan. *Error* terjadi karena adanya selisih dari data dengan target. Data dan target yang terdapat pada set data pun mempengaruhi kinerja dari fungsi *error* dalam menentukan *error* yang terjadi pada proses pelatihan. Jumlah epoh pun berpengaruh pada *error* minimum yang dicapai karena jumlah epoh menjadi batas dalam jumlah pengulangan proses pelatihan yang terjadi pada proses pelatihan dan batas *error* minimum menjadi batas untuk kondisi berhenti jika *error* yang dihasilkan dalam pelatihan telah mencapai *error* yang diinginkan.

BAB 4
JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN FUNGSI *ERROR CROSS-ENTROPY*

Pada bagian ini akan dijelaskan mengenai tujuan percobaan, algoritma, grafik hasil percobaan, dan analisis percobaan dari Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy*.

4.1 Tujuan Percobaan JST RBF dengan Fungsi *Error Cross-Entropy*

Pada percobaan yang dilakukan ingin dicapai tujuan sebagai berikut:

1. Memahami kinerja dari Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy* .
2. Mengetahui akibat-akibat dari perubahan parameter-parameter yang ada pada Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy* .

4.2 Prosedur Percobaan JST RBF dengan Fungsi *Error Cross-Entropy*

Proses yang dilakukan pada percobaan ini adalah mengumpulkan data, menyusun data, melakukan tahap *z-score* pada data agar lebih mudah untuk diolah oleh jaringan. Selanjutnya, data-data tersebut dilatih pada jaringan dengan rasio data 50% dari total tiap set data secara keseluruhan. Jaringan akan berhenti melakukan pelatihan apabila telah mencapai batas *error* minimum sebesar 0.01 atau epoh maksimal sebesar 10000. Setelah mengalami pelatihan, dilakukan pengujian pada data yang digunakan untuk pelatihan dan 50% set data yang tidak digunakan dalam pelatihan. Diagram alur algoritma jaringan dapat dilihat pada subbab 4.3. Untuk mendapatkan data dari hasil percobaan digunakan komputer dengan spesifikasi sebagai berikut:

Prosesor	: Intel Xeon T7100 @2,66 GHz
Memori	: 3328 MB RAM
Sistem Operasi	: Windows XP Professional SP 3
Perangkat Lunak	: MATLAB R2009a

Pada Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy* terjadi 2 kali proses propagasi yaitu propagasi maju dan propagasi balik. Untuk propagasi maju, proses yang terjadi adalah data masukan masuk ke jaringan melalui lapisan masukan dan langsung menuju ke lapisan tersembunyi. Di lapisan tersembunyi, data masukan dari setiap neuron lapisan masukan mengalami perhitungan antara data masukan dengan nilai tengah data masukan dan kemudian dengan fungsi aktifasi Gaussian.

$$Z_{in} = \frac{\sum (x - c)^2}{\sigma^2} \tag{4.1}$$

$$Z = e^{-Z_{in}} \tag{4.2}$$

Perhitungan keluaran dari lapisan tersembunyi ini dipengaruhi oleh bias serta bobot antara lapisan tersembunyi dengan lapisan keluaran. Lalu keluaran dari lapisan keluaran adalah hasil dari perhitungan fungsi sigmoid unipolar.

$$Y_{in} = w_{0l} + \sum_{m=1}^{M} Z \cdot w_{ml} \tag{4.3}$$

$$Y = \frac{1}{(1 + e^{-Y_{in}})} \tag{4.4}$$

Pada Jaringan Saraf Tiruan *Radial Basis Function* ini, besarnya kesalahan yang terjadi pada keluaran terhadap target yang bersesuaian dihitung setelah propagaasi maju selesai dengan menggunakan fungsi e*rror cross-entropy* seperti pada persamaan di bawah ini:

$$E = \frac{1}{2} \sum_{l=1}^{L} (T \ln(Y) - (1 - T) \ln(1 - Y))^2 \tag{4.5}$$

Pada saat propagasi balik terdapat koreksi nilai-nilai dari bobot dan bias dari lapisan tersembunyi ke lapisan keluaran serta nilai tengah dan lebar data. Koreksi dari nilai-nilai tersebut didapat dari turunan rumus *error* terhadap

komponen yang ingin dikoreksi. Berikut ini akan diperlihatkan turunan-turunan dari setiap fungsi yang akan digunakan pada proses propagasi balik.

1. Turunan fungsi *error* terhadap fungsi masukan pada lapisan keluaran:

$$\frac{\partial E}{\partial Y_{in}} = \frac{\partial E}{\partial Y}\frac{\partial Y}{\partial Y_{in}} = -(T - Y) \tag{4.6}$$

2. Turunan fungsi masukan dari lapisan keluaran terhadap fungsi keluaran pada lapisan tersembunyi:

$$\frac{\partial Y_{in}}{\partial Z} = \sum_{m=1}^{M} w_{ml} \tag{4.7}$$

3. Turunan fungsi keluaran dari lapisan tersembunyi terhadap fungsi masukan pada lapisan tersembunyi dan persamaan (4.2) disubtitusi ke dalam hasil turunan, maka didapat:

$$\frac{\partial Z}{\partial Z_{in}} = -e^{-Z_{in}} = -Z \tag{4.8}$$

4. Turunan fungsi masukan dari lapisan tersembunyi terhadap nilai tengah:

$$\frac{\partial Z_{in}}{\partial c} = \frac{-\sum(x - c)}{\sigma^2} \tag{4.9}$$

5. Turunan fungsi masukan dari lapisan tersembunyi terhadap nilai lebar data:

$$\frac{\partial Z_{in}}{\partial \sigma} = \frac{\sum(x - c)^2}{\sigma^3} \tag{4.10}$$

6. Turunan fungsi masukan dari lapisan tersembunyi terhadap bobot antara lapisan tersembunyi dengan lapisan keluaran:

$$\frac{\partial Y_{in}}{\partial w_{ml}} = Z \tag{4.11}$$

Berdasarkan turunan rumus masing-masing bagian yang telah diuraikan, maka turunan rumus untuk koreksi nilai bobot terhadap *error* sebagai berikut:

$$\frac{\partial E}{\partial w_{ml}} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial w_{ml}}$$

$$\frac{\partial E}{\partial w_{ml}} = -(T - Y) \times Z \tag{4.12}$$

Untuk koreksi nilai tengah, maka turunan rumus terhadap *error* sebagai berikut:

$$\frac{\partial E}{\partial c} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial Z} \times \frac{\partial Z}{\partial Z_{in}} \times \frac{\partial Z_{in}}{\partial c}$$

$$\frac{\partial E}{\partial c} = -(T - Y) \times \sum_{m=1}^{M} w_{ml} \times -Z \times \frac{\sum (x - c)}{\sigma^2} \tag{4.13}$$

Untuk koreksi nilai lebar data, maka turunan rumus terhadap *error* sebagai berikut:

$$\frac{\partial E}{\partial \sigma} = \frac{\partial E}{\partial Y} \times \frac{\partial Y}{\partial Y_{in}} \times \frac{\partial Y_{in}}{\partial Z} \times \frac{\partial Z}{\partial Z_{in}} \times \frac{\partial Z_{in}}{\partial \sigma}$$

$$\frac{\partial E}{\partial \sigma} = -(T - Y) \times \sum_{m=1}^{M} w_{ml} \times -Z \times \frac{\sum (x - c)^2}{\sigma^3} \tag{4.14}$$

4.3 Algoritma Percobaan JST RBF dengan Fungsi *Error Cross-Entropy*

Algoritma yang digunakan dalam melakukan pemrograman jaringan saraf tiruan dengan algoritma *Radial Basis Function* dengan Fungsi *Error Cross-Entropy* menggunakan algortima sebagai berikut:

1. Inisialisasi
    a) Inisialisasi bobot dan bias untuk lapisan tersembunyi ke lapisan keluaran dengan menggunakan metode Nguyen-Widrow.
    b) Menentukan vektor masukan dan vektor target, vektor nilai tengah RBF serta lebar data RBF.

2. Jika kondisi berhenti belum terpenuhi, maka jalankan langkah 3-5.

3. Melakukan proses propagasi maju
   a) Setiap unit *input* menerima sinyal masukan dan menyebarkan sinyal tersebut ke lapisan tersembunyi.
   b) Setiap unit tersembunyi mengalami perhitungan terhadap vektor nilai tengah.

$$Z_{in} = \frac{\sum (x - c)^2}{\sigma^2} \tag{4.15}$$

   c) Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit tersembunyi.

$$Z = e^{-Z_{in}} \tag{4.16}$$

   d) Setiap unit keluaran menjumlahkan sinyal masukan yang telah diberi bobot.

$$Y_{in} = w_{0l} + \sum_{m=1}^{M} Z \cdot w_{ml} \tag{4.17}$$

   e) Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran dari unit keluaran

$$Y = \frac{1}{(1 + e^{-Y_{in}})} \tag{4.18}$$

4. Melakukan propagasi balik
   a) Setiap unit keluaran menerima sebuah pola target yang bersesuaian dengan pola pelatihan masukan lalu menghitung informasi kesalahan

$$\delta = -(T - Y) \tag{4.19}$$

b) Menghitung koreksi bobotnya dan menghitung koreksi biasnya

$$\Delta w_{ml} = \alpha \cdot \delta \cdot Z \tag{4.20}$$

$$\Delta w_{0l} = \alpha \cdot \delta \tag{4.21}$$

c) Menghitung koreksi nilai tengah dan lebar data RBF dengan cara menggunakan:

$$\Delta c = \beta \cdot \delta \cdot \left(\sum_{m=1}^{M} w_{ml} \right) (-Z) \frac{-\sum (x - c)}{\sigma^2} \tag{4.22}$$

Untuk menghitung koreksi nilai lebar data RBF, dilakukan perhitungan awal pada setiap kelas pada set data masukan (1 sampai dengan n) dengan persamaan:

$$\Delta \sigma_n = \gamma \cdot \delta \cdot \left(\sum_{m=1}^{M} w_{ml} \right) (-Z) \frac{\sum (x - c)^2}{\sigma^3} \tag{4.23}$$

Setelah itu, maka dihitung nilai rata-rata dari Δσ menggunakan persamaan:

$$\Delta \sigma = \Delta \sigma_1 + \Delta \sigma_2 + ... + \Delta \sigma_n \tag{4.24}$$

5. Pengubahan nilai-nilai bobot, bias, nilai tengah, dan lebar RBF dengan persamaan:

$$w_{ml} = w_{ml} + \Delta w_{ml} \tag{4.25}$$

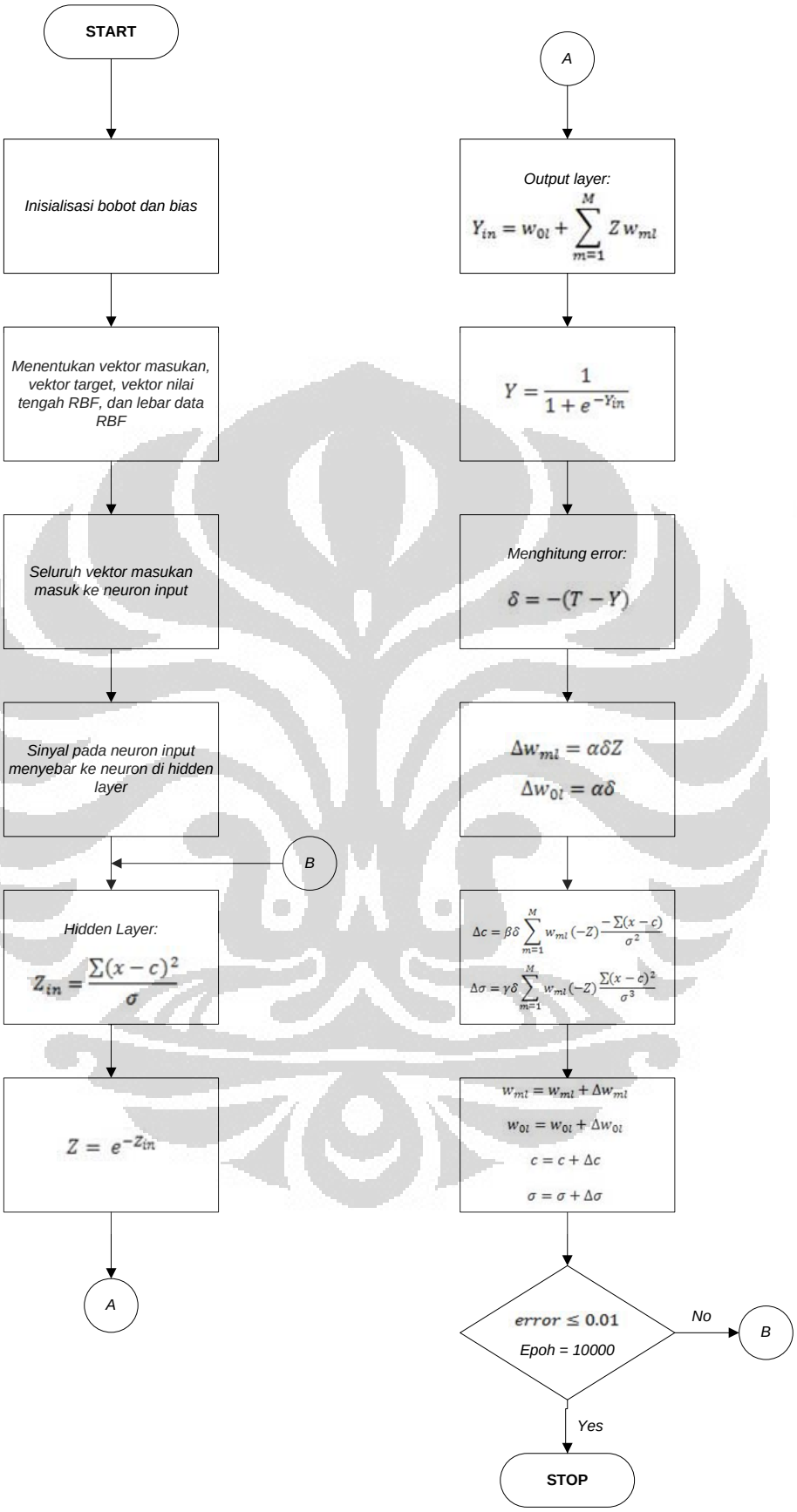
$$w_{0l} = w_{0l} + \Delta w_{0l} \tag{4.26}$$

$$c = c + \Delta c \tag{4.27}$$

$$\sigma = \sigma + \Delta \sigma \tag{4.28}$$

6. Pengujian kondisi berhenti.

   Berhenti jika *error* telah mencapai 0.01 atau epoh telah mencapai 10000.

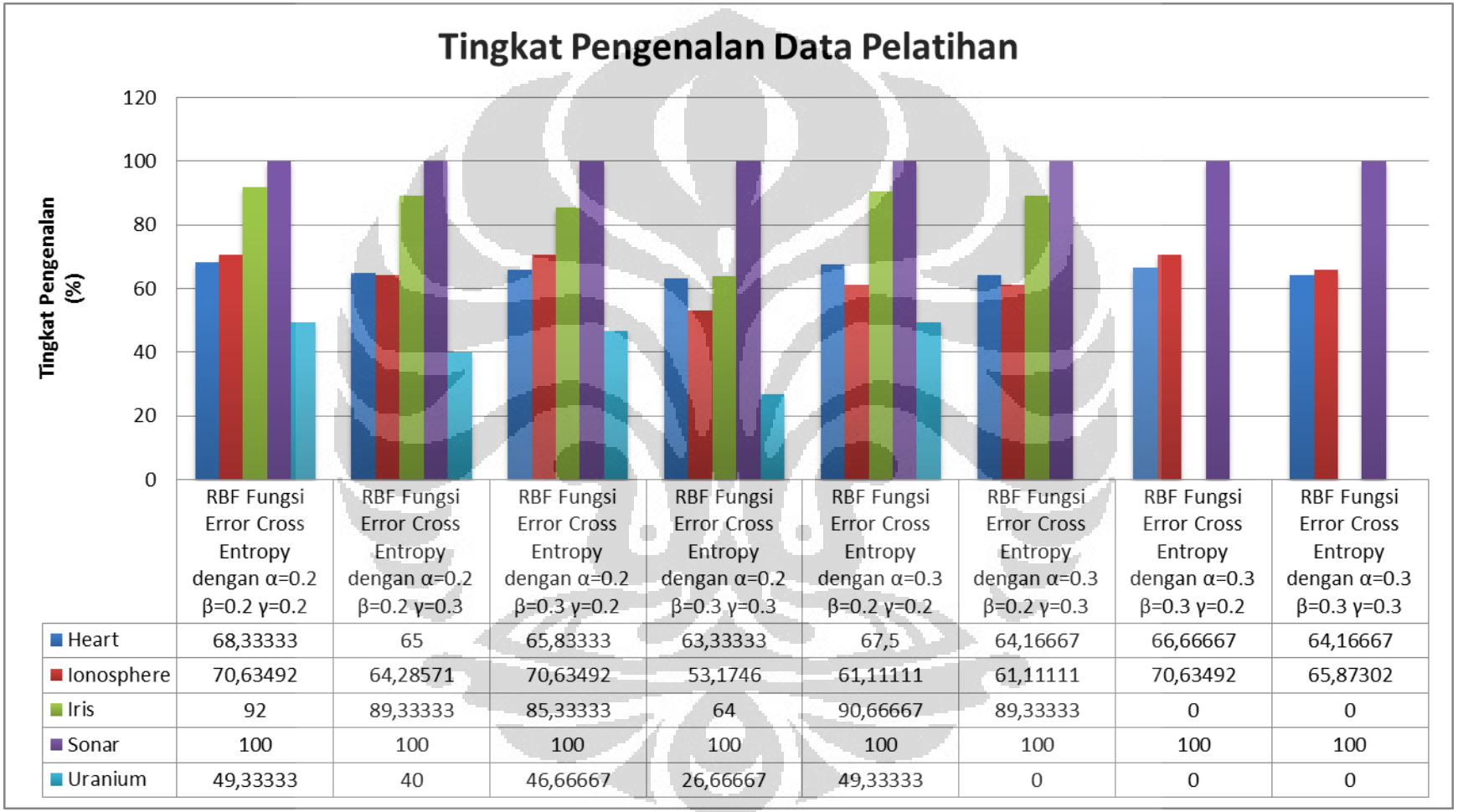


Gambar 4.1 Diagram Blok Algoritma JST RBF Fungsi *Error Cross-Entropy*

4.4 Hasil Percobaan JST RBF dengan Fungsi *Error Cross-Entropy*

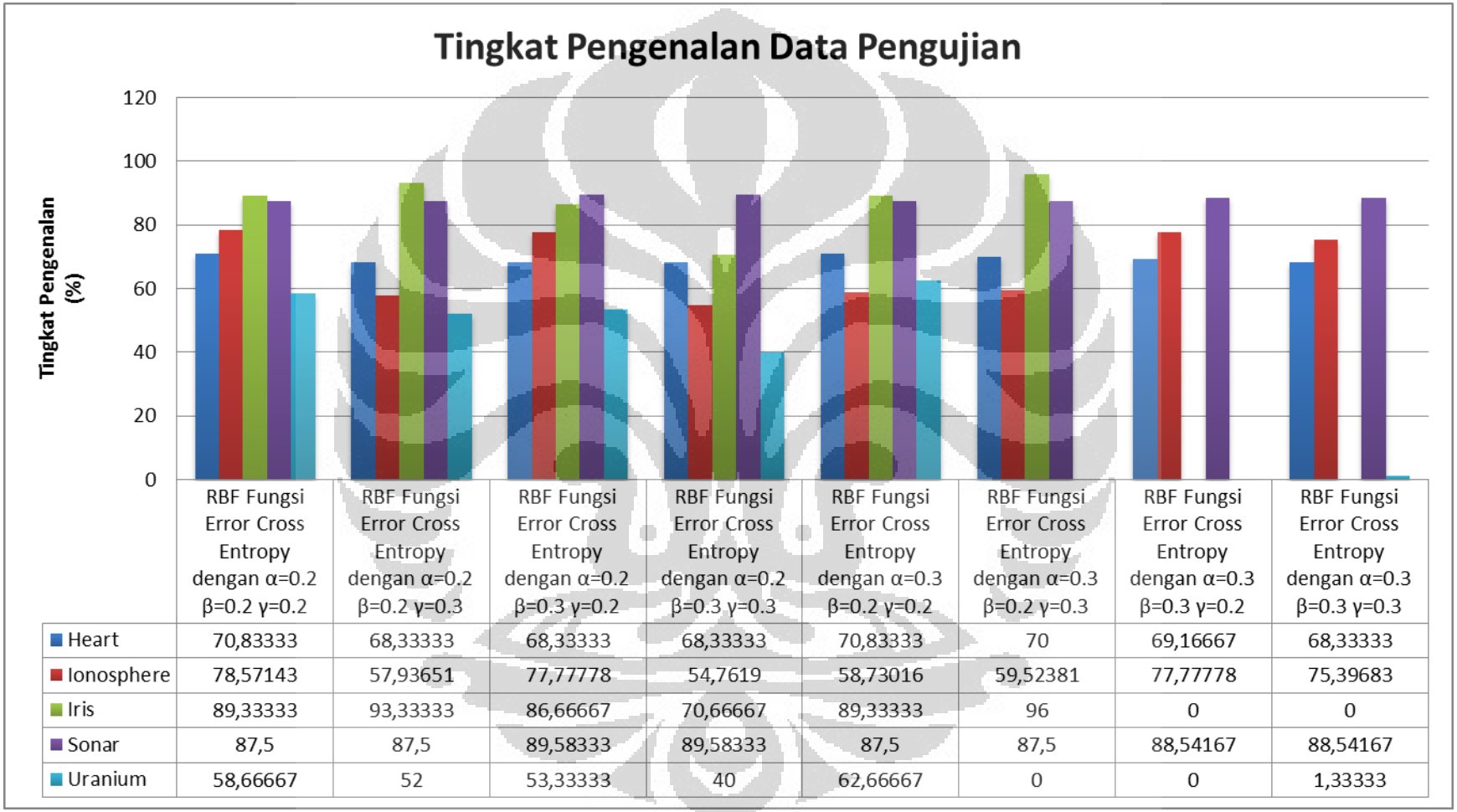
Pada bagian ini akan diperlihatkan grafik-grafik hasil pengolahan data yang didapat dari Jaringan Saraf Tiruan *Radial Basis Function* dengan Fungsi *Error Cross-Entropy* . Parameter-parameter yang digunakan untuk mendapatkan hasil percobaan tersebut adalah sebagai berikut:

Rasio data untuk pelatihan (dalam %)	= 50
Variasi konstanta pembelajaran (α)	= 0.2 dan 0.3
Variasi konstanta pembelajaran nilai tengah (β)	= 0.2 dan 0.3
Variasi konstanta pembelajaran lebar data (γ)	= 0.2 dan 0.3
Kondisi stop *error* minimum	= 0.01
Kondisi stop epoh maksimum	= 10000



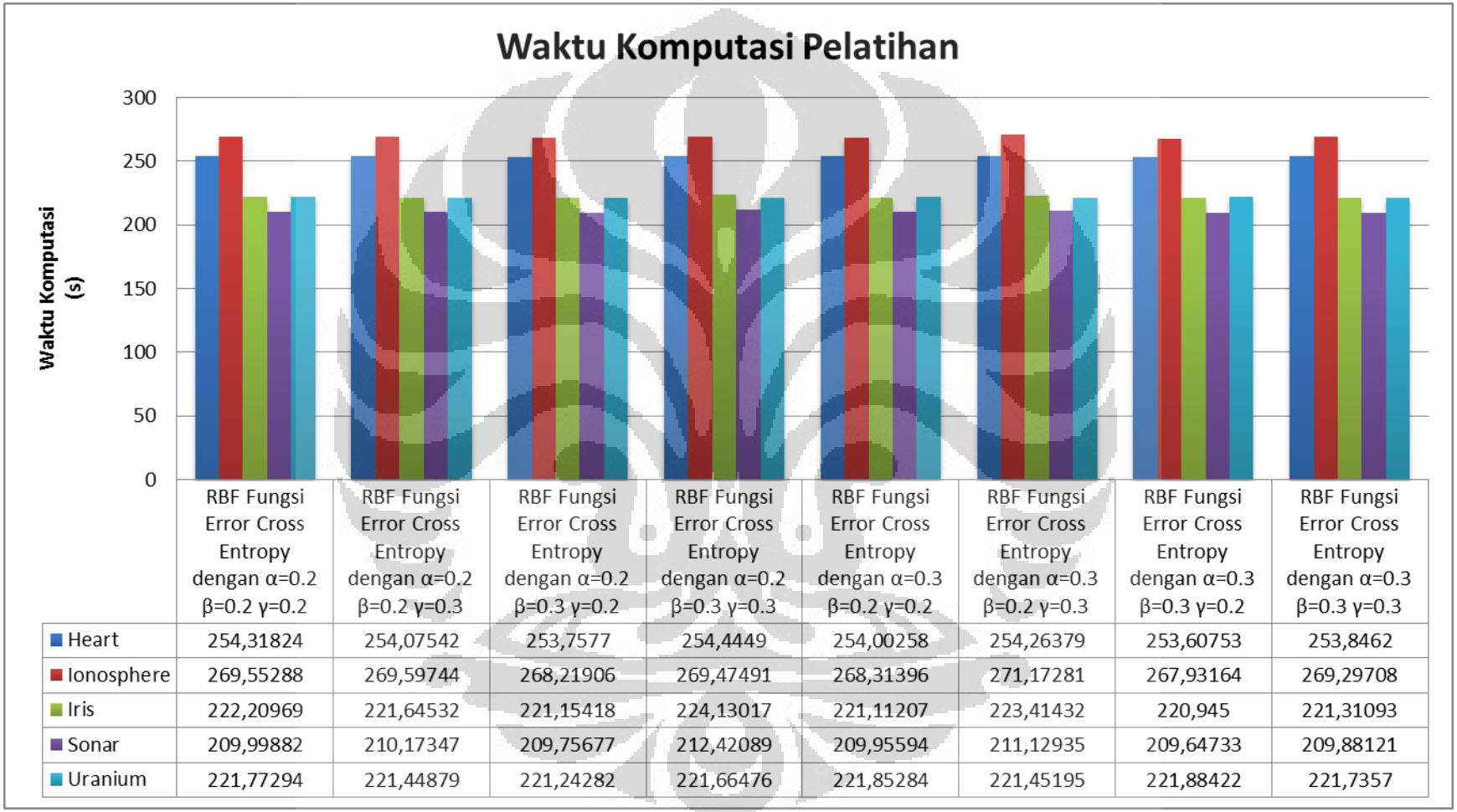
	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	68,33333	65	65,83333	63,33333	67,5	64,16667	66,66667	64,16667
Ionosphere	70,63492	64,28571	70,63492	53,1746	61,11111	61,11111	70,63492	65,87302
Iris	92	89,33333	85,33333	64	90,66667	89,33333	0	0
Sonar	100	100	100	100	100	100	100	100
Uranium	49,33333	40	46,66667	26,66667	49,33333	0	0	0

Gambar 4.2 Tingkat Pengenalan Data Pelatihan pada JST RBF Fungsi *Error Cross-Entropy*



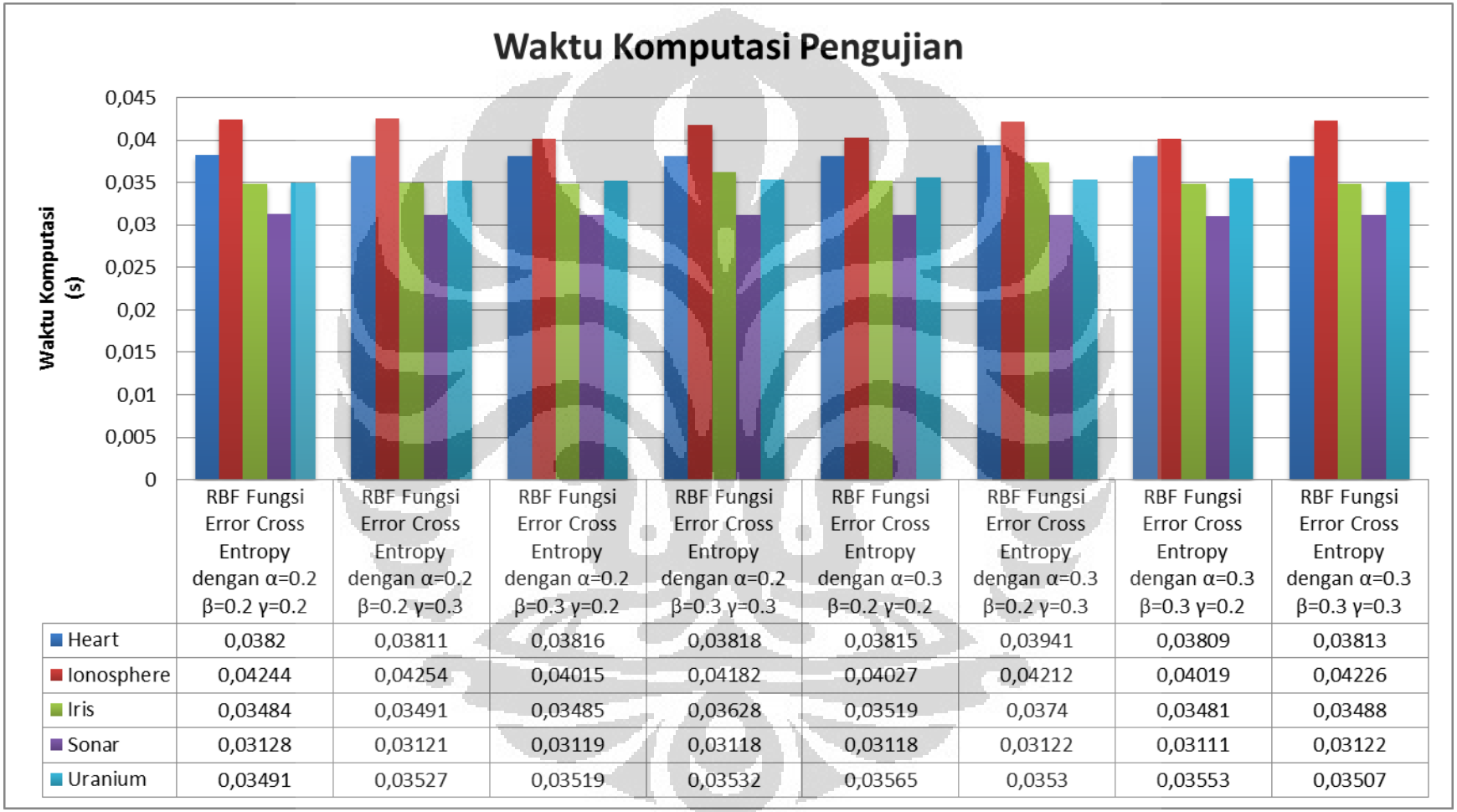
	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	70,83333	68,33333	68,33333	68,33333	70,83333	70	69,16667	68,33333
Ionosphere	78,57143	57,93651	77,77778	54,7619	58,73016	59,52381	77,77778	75,39683
Iris	89,33333	93,33333	86,66667	70,66667	89,33333	96	0	0
Sonar	87,5	87,5	89,58333	89,58333	87,5	87,5	88,54167	88,54167
Uranium	58,66667	52	53,33333	40	62,66667	0	0	1,33333

Gambar 4.3 Tingkat Pengenalan Data Pengujian pada JST RBF Fungsi *Error Cross-Entropy*



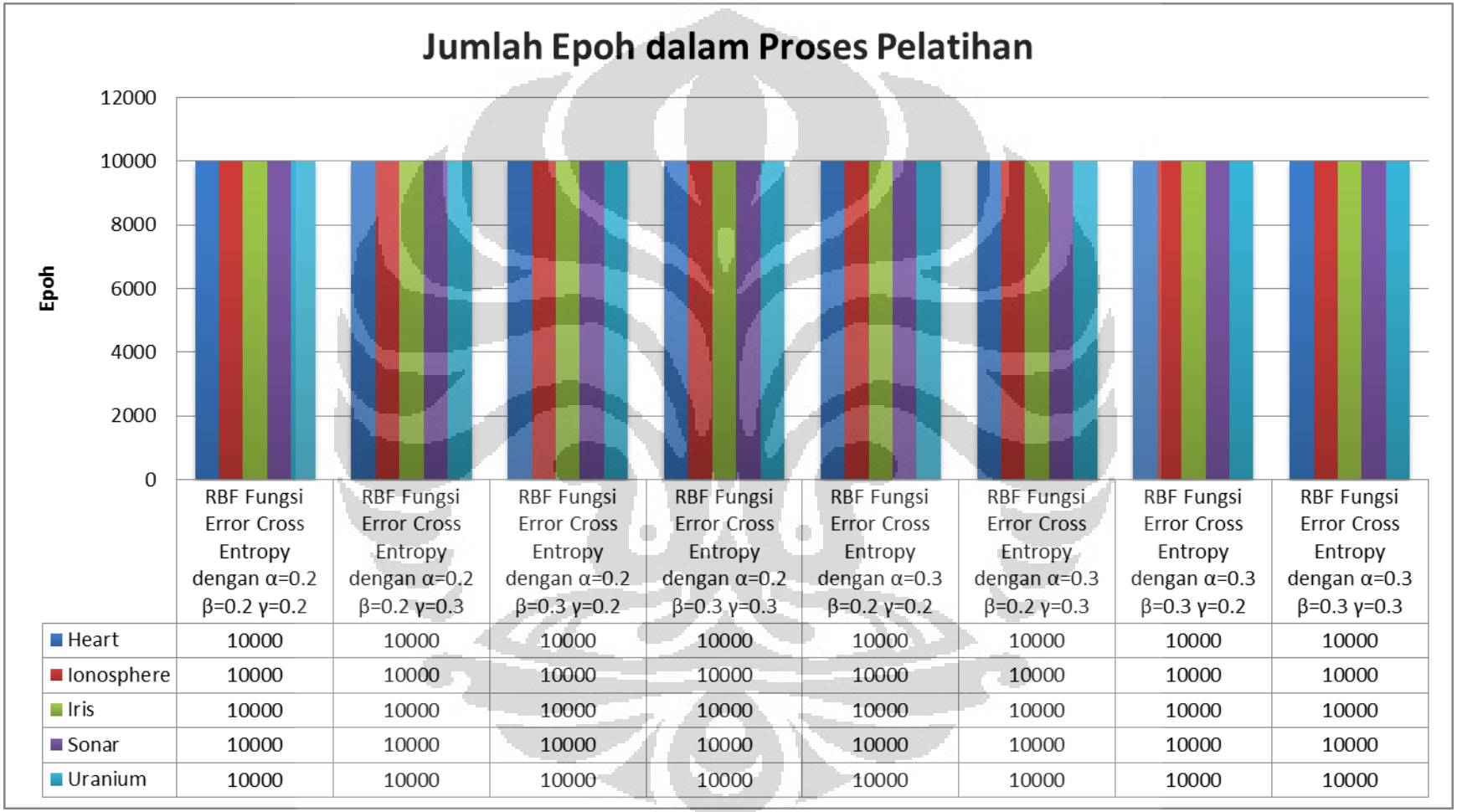
	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	254,31824	254,07542	253,7577	254,4449	254,00258	254,26379	253,60753	253,8462
Ionosphere	269,55288	269,59744	268,21906	269,47491	268,31396	271,17281	267,93164	269,29708
Iris	222,20969	221,64532	221,15418	224,13017	221,11207	223,41432	220,945	221,31093
Sonar	209,99882	210,17347	209,75677	212,42089	209,95594	211,12935	209,64733	209,88121
Uranium	221,77294	221,44879	221,24282	221,66476	221,85284	221,45195	221,88422	221,7357

Gambar 4.4 Waktu Komputasi Pelatihan pada JST RBF Fungsi *Error Cross-Entropy*



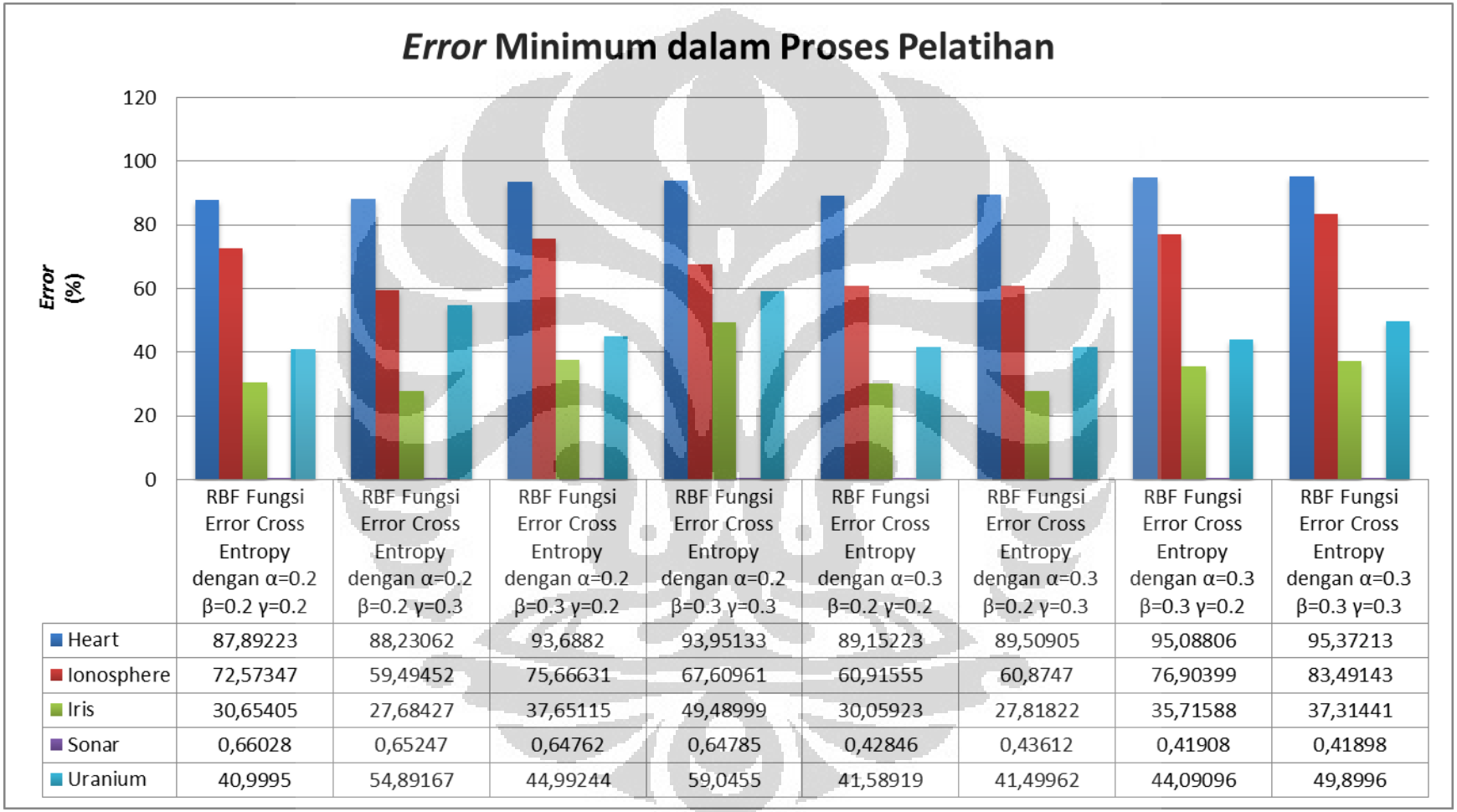
	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	0,0382	0,03811	0,03816	0,03818	0,03815	0,03941	0,03809	0,03813
Ionosphere	0,04244	0,04254	0,04015	0,04182	0,04027	0,04212	0,04019	0,04226
Iris	0,03484	0,03491	0,03485	0,03628	0,03519	0,0374	0,03481	0,03488
Sonar	0,03128	0,03121	0,03119	0,03118	0,03118	0,03122	0,03111	0,03122
Uranium	0,03491	0,03527	0,03519	0,03532	0,03565	0,0353	0,03553	0,03507

Gambar 4.5 Waktu Komputasi Pengujian pada JST RBF Fungsi *Error Cross-Entropy*



	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	10000	10000	10000	10000	10000	10000	10000	10000
Ionosphere	10000	10000	10000	10000	10000	10000	10000	10000
Iris	10000	10000	10000	10000	10000	10000	10000	10000
Sonar	10000	10000	10000	10000	10000	10000	10000	10000
Uranium	10000	10000	10000	10000	10000	10000	10000	10000

Gambar 4.6 Jumlah Epoh dalam Proses Pelatihan pada JST RBF Fungsi *Error Cross-Entropy*



	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	87,89223	88,23062	93,6882	93,95133	89,15223	89,50905	95,08806	95,37213
Ionosphere	72,57347	59,49452	75,66631	67,60961	60,91555	60,8747	76,90399	83,49143
Iris	30,65405	27,68427	37,65115	49,48999	30,05923	27,81822	35,71588	37,31441
Sonar	0,66028	0,65247	0,64762	0,64785	0,42846	0,43612	0,41908	0,41898
Uranium	40,9995	54,89167	44,99244	59,0455	41,58919	41,49962	44,09096	49,8996

Gambar 4.7 *Error* Minimum dalam Proses Pelatihan pada JST RBF Fungsi *Error Cross-Entropy*

4.5 Analisis Percobaan JST RBF dengan Fungsi *Error Cross-Entropy*

Pada percobaan kali ini akan diamati hasil dari jaringan saraf tiruan *Radial Basis Function* dengan fungsi *error cross-entropy*. Data yang diamati pada bagian tingkat pengenalan data pelatihan dan pengujian, waktu komputasi pelatihan dan pengujian, jumlah epoh dalam proses pelatihan, serta *error* minimum yang dicapai dalam proses pelatihan. Pada tingkat pengenalan data pelatihan didapatkan hasil yang baik untuk set data *sonar*, yakni mencapai 100% untuk seluruh percobaan. Sedangkan untuk set data *heart* memiliki tingkat pengenalan di bawah 70% pada seluruh percobaan. Hampir semua set data *ionosphere* memiliki tingkat pengenalan di bawah 70% kecuali pada percobaan dengan α, β, dan γ yang bernilai 0.2 serta α dan β yang bernilai 0.2 dan γ bernilai 0.3. Pada set data *iris* dan uranium terdapat tingkat pengenalan 0%. Pada tingkat pengenalan data pengujian terjadi hal yang sama seperti pada tingkat pengenalan data pelatihan, tetapi nilainya lebih rendah dari pada tingkat pengenalan data pelatihan. Tingkat pengenalan set data *iris* dan uranium sebesar 0%. Hasil tingkat pengenalan data pengujian terbaik terdapat pada set data *iris* pada percobaan RBF Fungsi *Error Cross-Entropy* dengan $\alpha = 0.3$; $\beta = 0{,}2$; dan $\gamma = 0.3$ yang mencapai 96%.

Waktu komputasi terbagi 2, yaitu waktu komputasi pelatihan dan waktu komputasi pengujian. Waktu komputasi pelatihan merupakan waktu yang digunakan untuk memproses komputasi pelatihan pada jaringan. Sedangkan waktu komputasi pengujian merupakan waktu yang digunakan untuk memproses komputasi pengujian pada jaringan. Keduanya memiliki waktu yang berbeda karena pada waktu komputasi pelatihan proses yang terjadi pada jaringan adalah proses propagasi maju dan propagsi balik sehingga memakan waktu yang lebih lama dibandingkan dengan waktu komputasi pengujian yang hanya melakukan proses komputasi maju diikuti penyeleksian data yang sesuai dengan target. Set data *sonar* memiliki waktu komputasi pelatihan yang tercepat di antara kelima set data dengan waktu komputasi rata-rata sebesar 210 detik. Set data yang memiliki waktu komputasi pelatihan terlama adalah set data *ionosphere* karena memiliki 252 data untuk tiap dimensi. Untuk waktu komputasi pengujian yang tercepat terdapat pada set data *sonar* dengan besar nilai rata-rata 0.031 dan waktu

komputasi terlama ada pada set data *ionosphere* dengan waktu komputasi rata-rata sebesar 0.042. Perbedaan waktu komputasi yang berbeda untuk setiap set data dikarenakan terdapat beberapa faktor yang mempengaruhi seperti jumlah kelas dan jumlah data yang diolah dari tiap set data. Kedua faktor tersebut akan mempengaruhi jumlah neuron pada lapisan tersembunyi. Neuron-neuron pada lapisan tersembunyi memiliki fungsi untuk menghitung jarak antara satu data terhadap nilai tengah dan lebar data untuk setiap kelas sehingga menyebabkan banyak terjadinya perhitungan aritmatika yang mempengaruhi lamanya komputasi. Selain itu, jumlah epoh dan batas minimum *error* pun mempengaruhi waktu komputasi karena jumlah epoh berguna sebagai batas dari jumlah pengulangan proses pelatihan yang terjadi pada percobaan. Batas minimum *error* juga berguna sebagai batas untuk kondisi berhenti dari jaringan jika telah mencapai *error* yang diinginkan. Apabila digunakan jumlah epoh yang berbeda-beda untuk melakukan percobaan, maka akan diperoleh waktu komputasi yang berbeda-beda pula. Pada percobaan ini, jumlah epoh yang digunakan sebagai batas maksimum pengulangan proses pelatihan sebesar 10000 epoh sehingga jaringan akan berhenti jika telah mencapai batas epoh tersebut.

Selain hal-hal di atas yang telah disebutkan, pada percobaan didapatkan juga *error* minimum. Nilai e*rror* minimum yang terkecil terdapat pada set data *sonar* dengan nilai rata-rata 0.38% sedangkan nilai *error* minimum yang terbesar terdapat pada set data *heart* dengan nilai rata-rata 11%. Pada pencapaian *error* minimum, fungsi *error* yang digunakan pada proses pelatihan sangat berpengaruh karena perhitungan fungsi *error* akan menghasilkan *error* antara data dengan target yang diharapkan. *Error* terjadi karena adanya selisih dari data dengan target sehingga data dan target yang terdapat pada set data pun mempengaruhi kinerja dari fungsi *error* dalam menentukan *error* yang terjadi pada proses pelatihan. Jumlah epoh pun berpengaruh pada *error* minimum yang dicapai karena jumlah epoh menjadi batas dalam jumlah iterasi yang terjadi pada proses pelatihan serta batas *error* minimum menjadi batas untuk kondisi berhenti jika *error* yang dihasilkan dalam pelatihan telah mencapai *error* yang diinginkan.

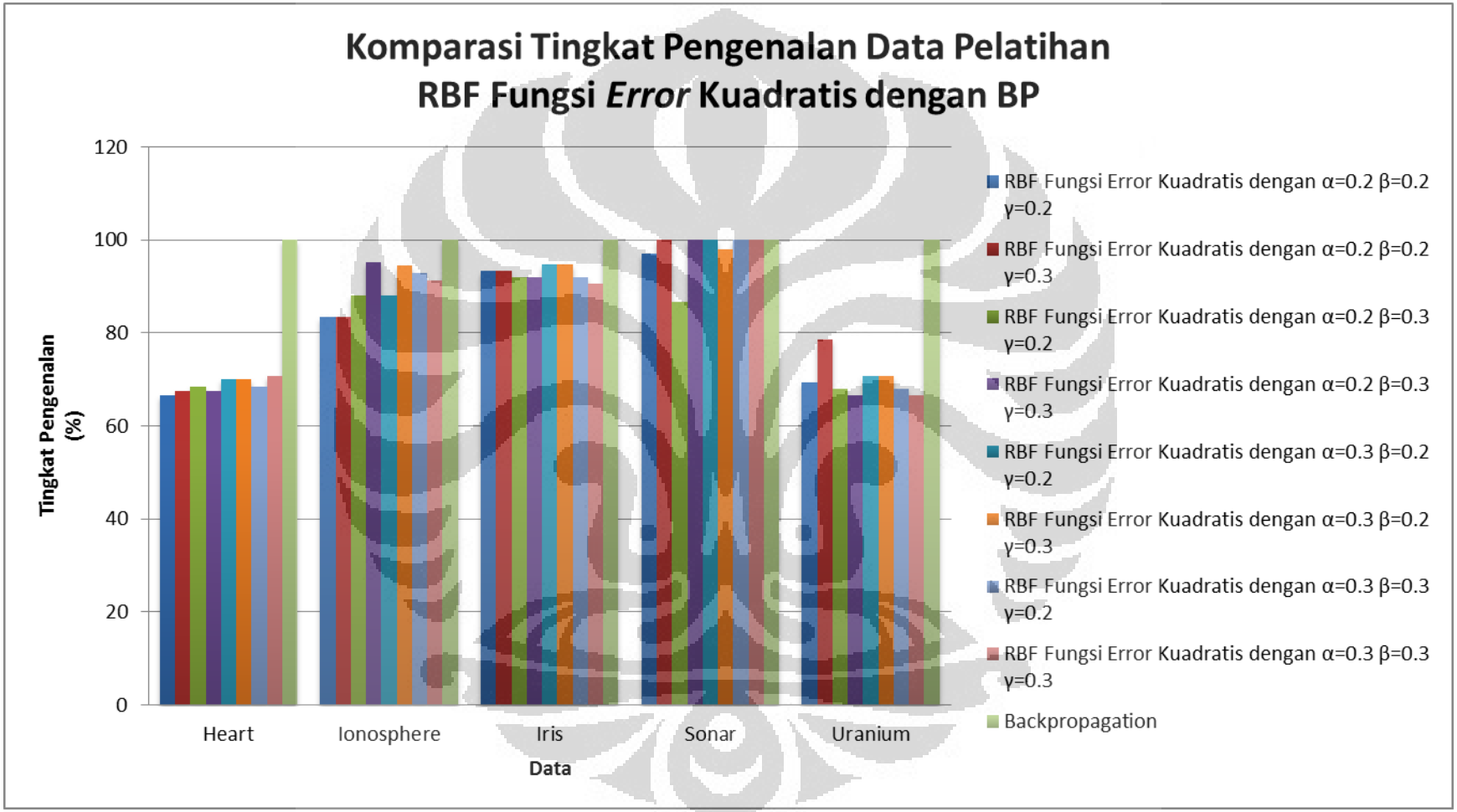
BAB 5
KOMPARASI JARINGAN SARAF TIRUAN ALGORITMA *RADIAL BASIS FUNCTION* DENGAN ALGORITMA *BACKPROPAGATION*

Pada bab ini akan dibahas mengenai perbandingan antara Jaringan Saraf Tiruan algoritma *Radial Basis Function* Fungsi *Error* Kuadratis dengan algoritma *Backpropagation,* algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy* dengan algoritma *Backpropagation*, dan algoritma *Radial Basis Function* Fungsi *Error* Kuadratis dengan algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy*. Agar bersesuaian, maka untuk komparasi ini proses pengambilan data menggunakan komputer yang sama agar tidak ada perbedaan komputasi, serta menggunakan parameter-parameter yang sama seperti *alpha* (α), *beta* (β), dan *gamma* (γ) yang bernilai 0.2, batas epoh maksimum sebesar 10000, batas *error* minimum sebesar 0.01, serta rasio data yang digunakan untuk pelatihan sebesar 50%. Dalam pengambilan data dari Jaringan Saraf Tiruan *Radial Basis Function* dan *Backpropagation*, spesifikasi komputer yang digunakan adalah:

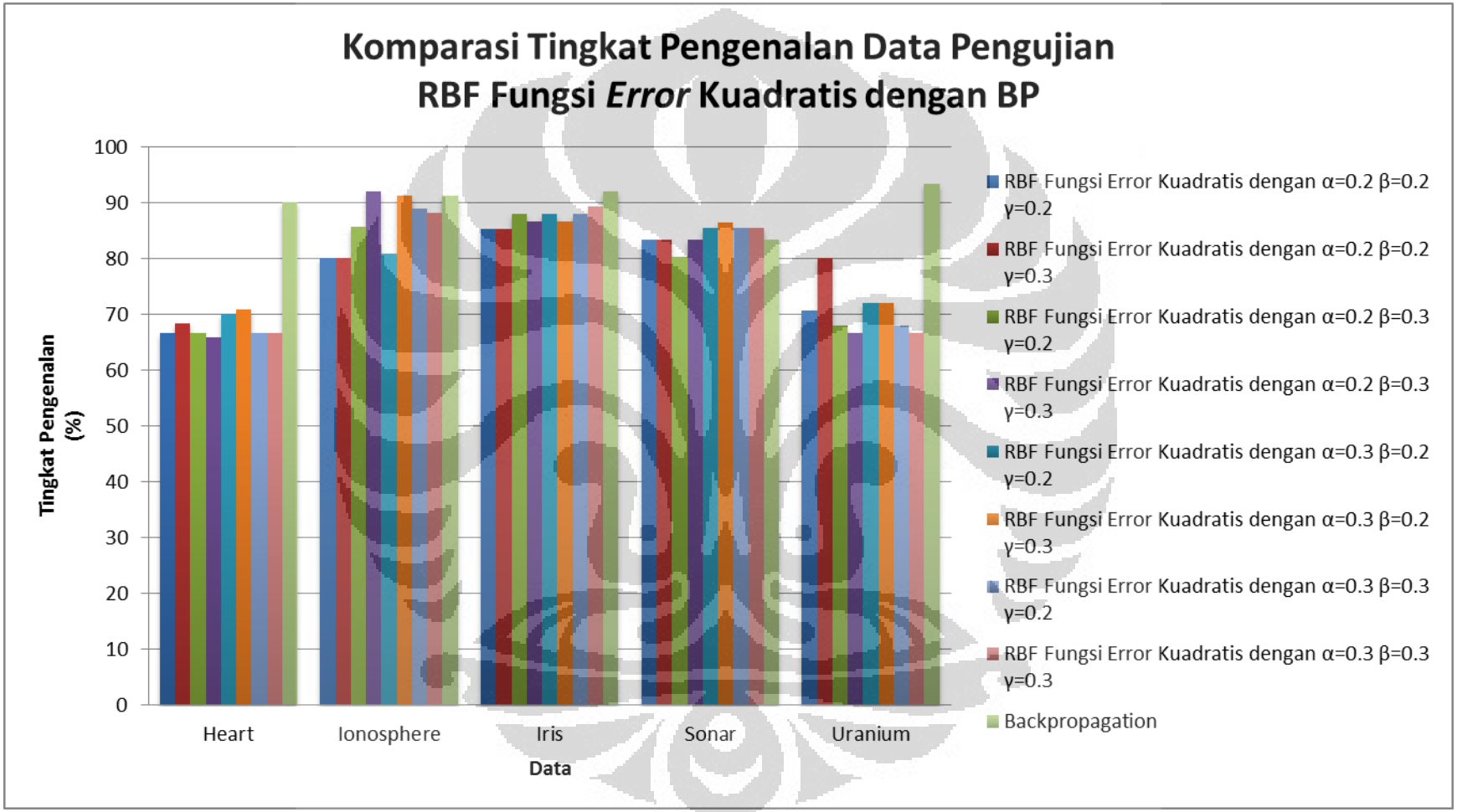
Prosesor	: Intel Xeon T7100 @2.66 GHz
Memori	: 3328 MB RAM
Sistem Operasi	: Windows XP Professional SP 3
Perangkat Lunak	: MATLAB R2009a

5.1 Komparasi Hasil Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Fungsi *Error* Kuadratis terhadap Algoritma *Backpropagation*

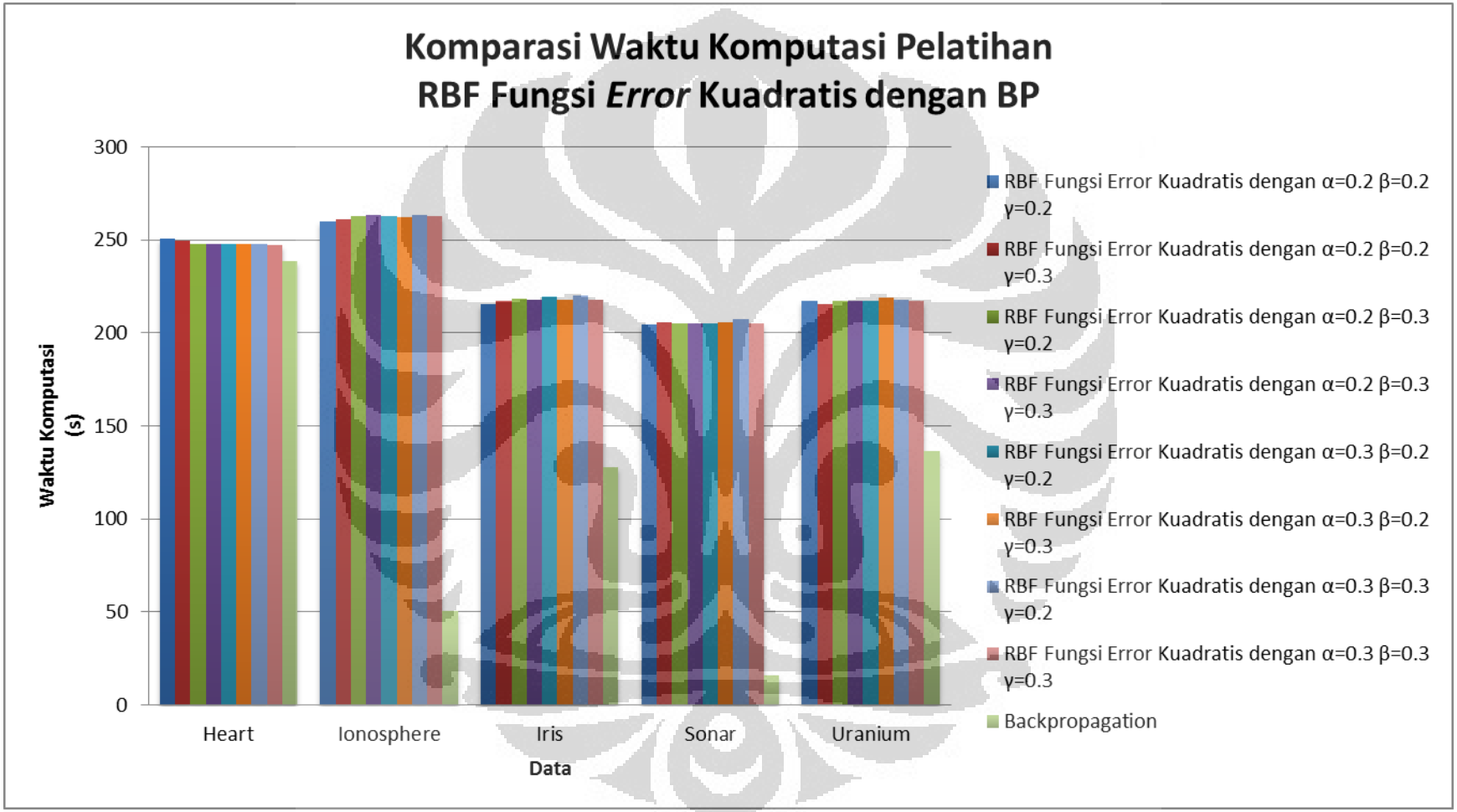
Pada komparasi ini terdapat batasan, yaitu data yang dikomparasi adalah data hasil dari Jaringan Saraf Tiruan *Radial Basis Function* (RBF) Fungsi *Error* Kuadratis dengan *Backpropagation* (BP) dengan parameter-parameter yang bersesuaian *alpha* (α), *beta* (β), dan *gamma* (γ) bernilai 0.2, batas epoh maksimum sebesar 10000, batas *error* minimum sebesar 0.01. Data yang digunakan adalah set data *heart*, *ionosphere*, *iris*, *sonar*, dan uranium. Komparasi yang dilakukan antara lain terhadap tingkat pengenalan data pelatihan dan data pengujian, waktu komputasi pelatihan dan pengujian, *error* minimum dalam proses pelatihan, serta epoh maksimum dalam proses pelatihan.



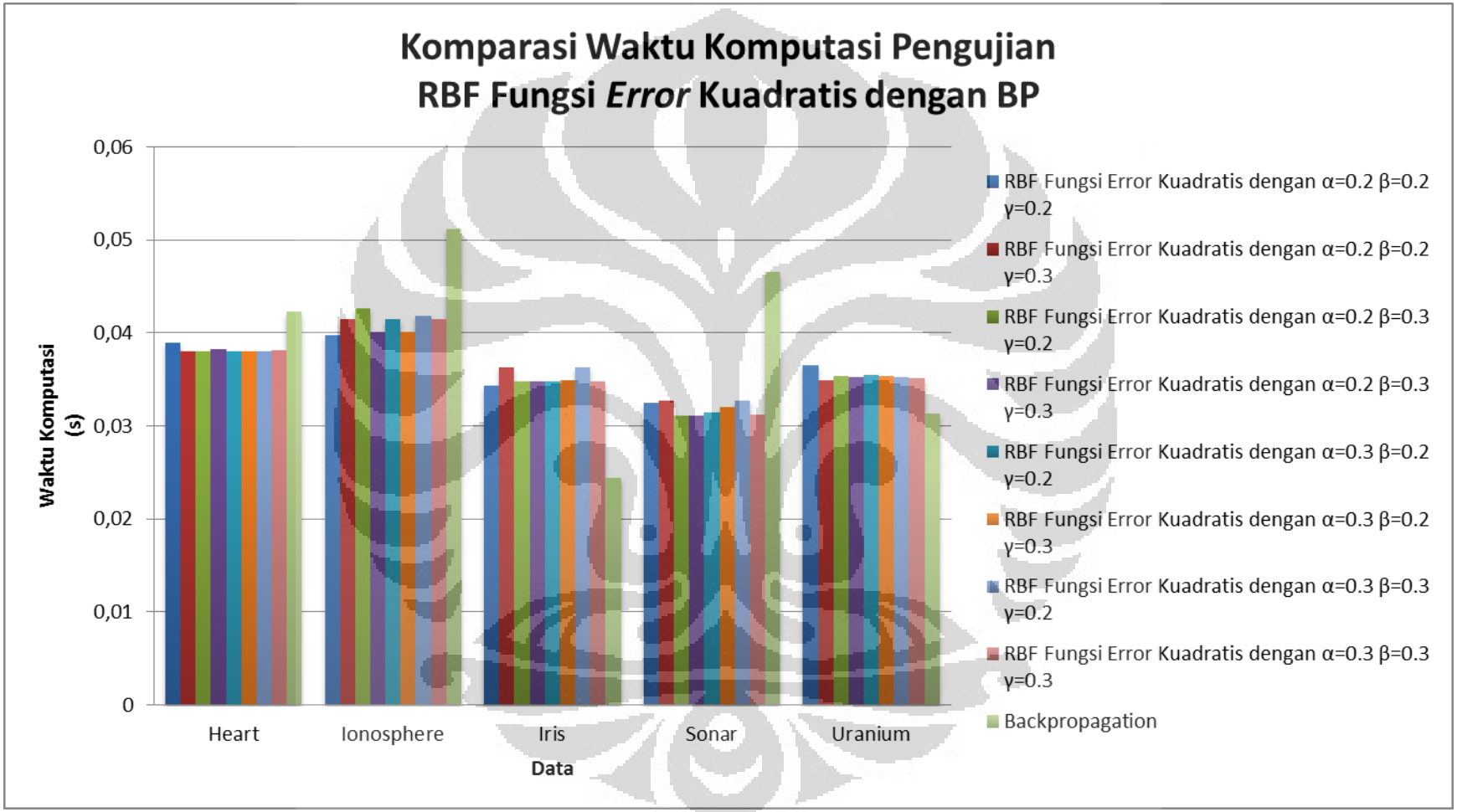
Gambar 5.1 Komparasi Tingkat Pengenalan Data Pelatihan RBF Fungsi *Error* Kuadratis dengan BP



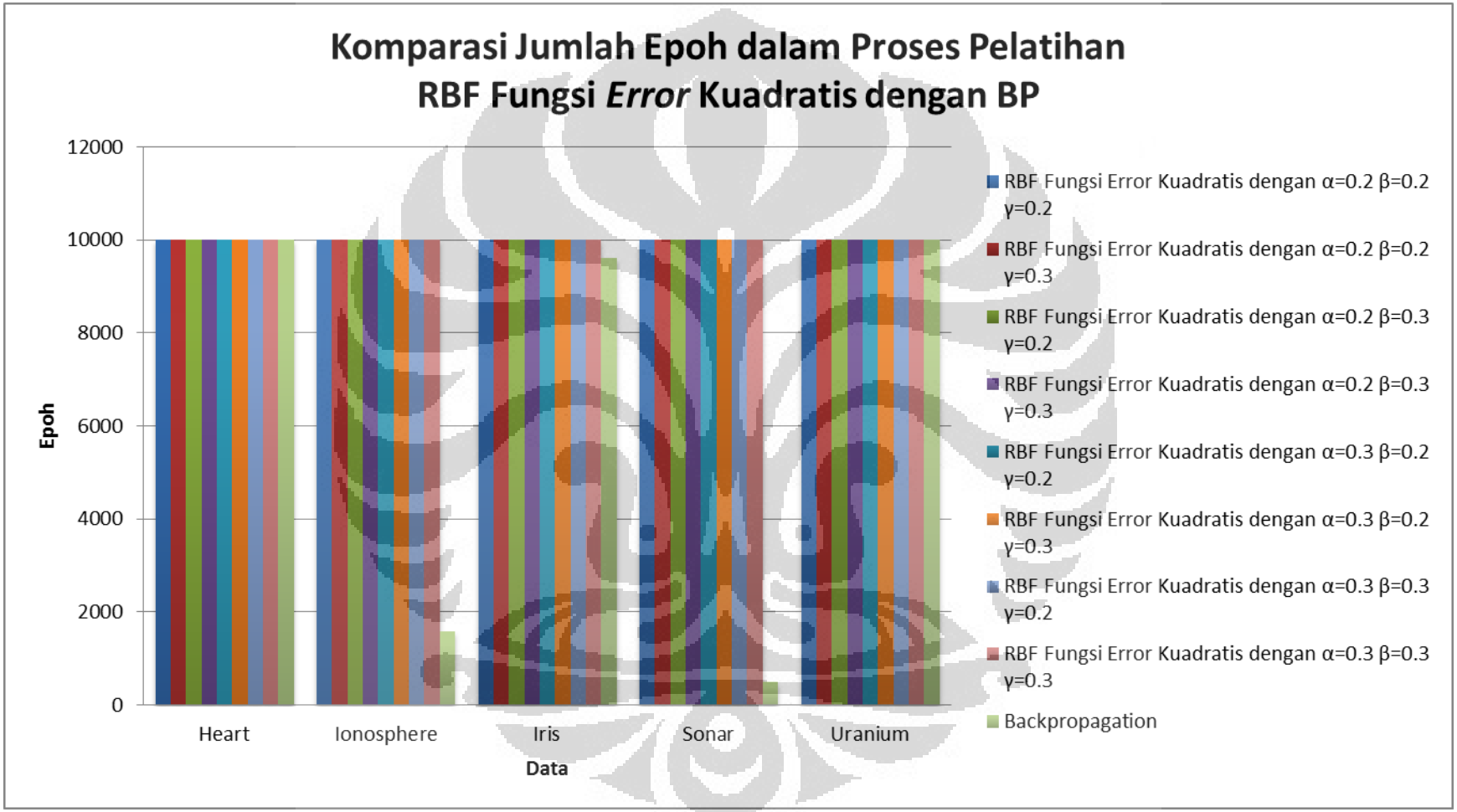
Gambar 5.2 Komparasi Tingkat Pengenalan Data Pengujian RBF Fungsi *Error* Kuadratis dengan BP



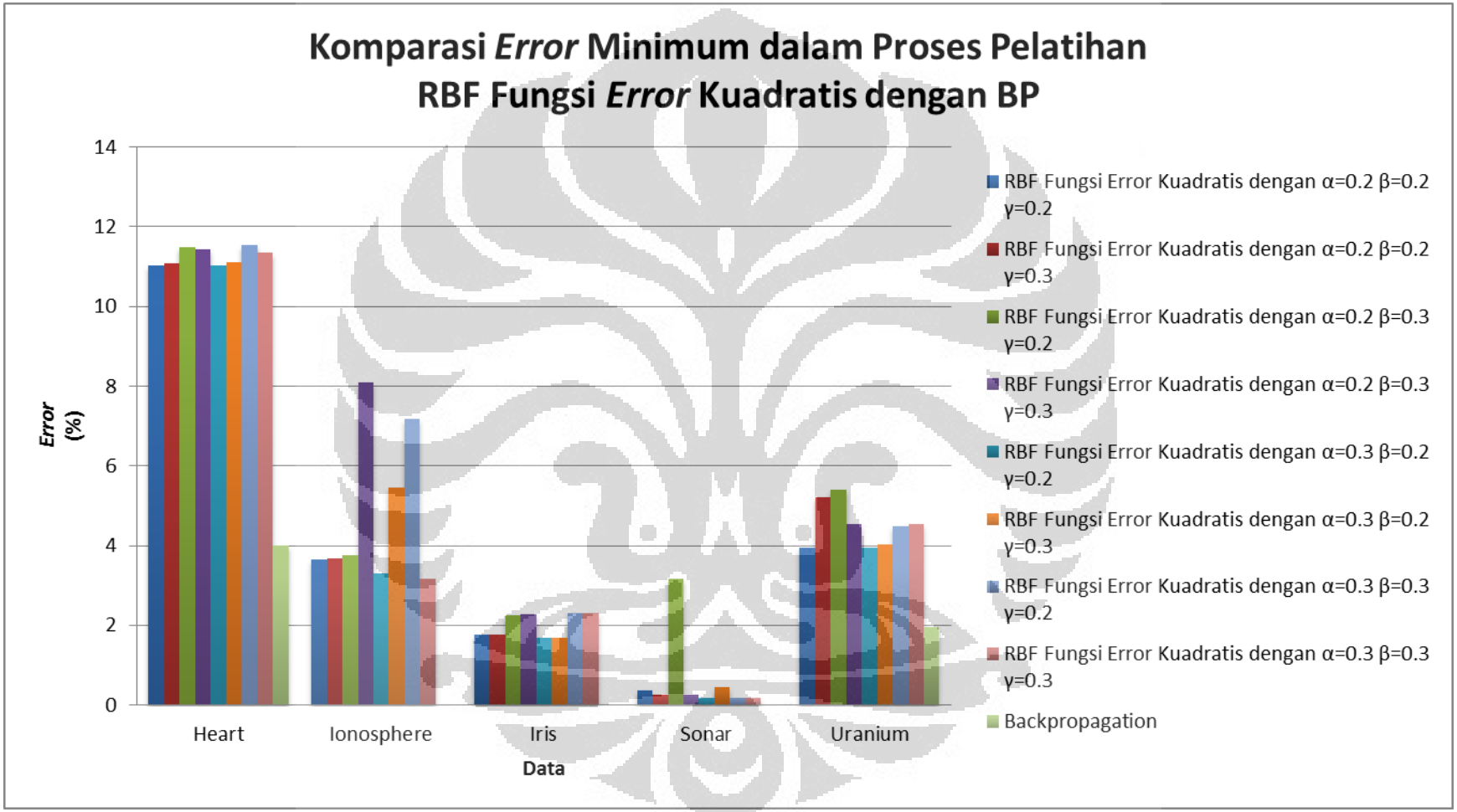
Gambar 5.3 Komparasi Waktu Komputasi Pelatihan RBF Fungsi *Error* Kuadratis dengan BP



Gambar 5.4 Komparasi Waktu Komputasi Pengujian RBF Fungsi *Error* Kuadratis dengan BP



Gambar 5.5 Komparasi Jumlah Epoh dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan BP



Gambar 5.6 Komparasi *Error* Minimum dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan BP

5.2 Analisis Komparasi Hasil Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Fungsi *Error* Kuadratis terhadap Algoritma *Backpropagation*

Subbab ini akan membahas mengenai komparasi dari hasil RBF Fungsi *Error* Kuadratis dengan *Backpropagation*. Hal yang dikomparasi antara lain adalah tingkat pengenalan data pelatihan dan data pengujian, waktu komputasi pelatihan dan pengujian, jumlah epoh dalam proses pelatihan, serta *error* minimum dalam proses pelatihan. Setelah mengomparasi untuk tingkat pengenalan data pelatihan dan pengujian, hasilnya menunjukkan bahwa hanya set data *iris* dan *sonar* yang dapat mendekati hasil dari algoritma *Backpropagation*. Set data *ionosphere* pun dapat mendekati, akan tetapi perbedaannya (selisih) mencapai 17% (hanya pada percobaan dengan *alpha* 0.2 *beta* 0.2 dan *gamma* 0.2 serta percobaan dengan *alpha* 0.2, *beta* 0.2, dan *gamma* 0.3) untuk data pelatihan dan 10% untuk data pengujian.

Komparasi waktu komputasi pelatihan menunjukkan bahwa *Backpropagation* lebih cepat daripada RBF Fungsi *Error* Kuadratis, akan tetapi itu terjadi karena pada algoritma *Backpropagation*, proses pelatihan berhenti karena *error*-nya telah mencapai kondisi berhenti. Jika proses pelatihan hanya dibatasi dengan jumlah epoh saja, maka waktu komputasi pada RBF Fungsi *Error* Kuadratis akan lebih cepat daripada algoritma *Backpropagation*. Hal ini dapat dilihat pada waktu komputasi pengujian, waktu komputasi pengujian untuk set data *heart*, *ionosphere*, dan *sonar* lebih cepat daripada dengan algoritma *Backpropagation*. Namun, pada set data *iris* dan *uranium* justru terjadi sebaliknya.

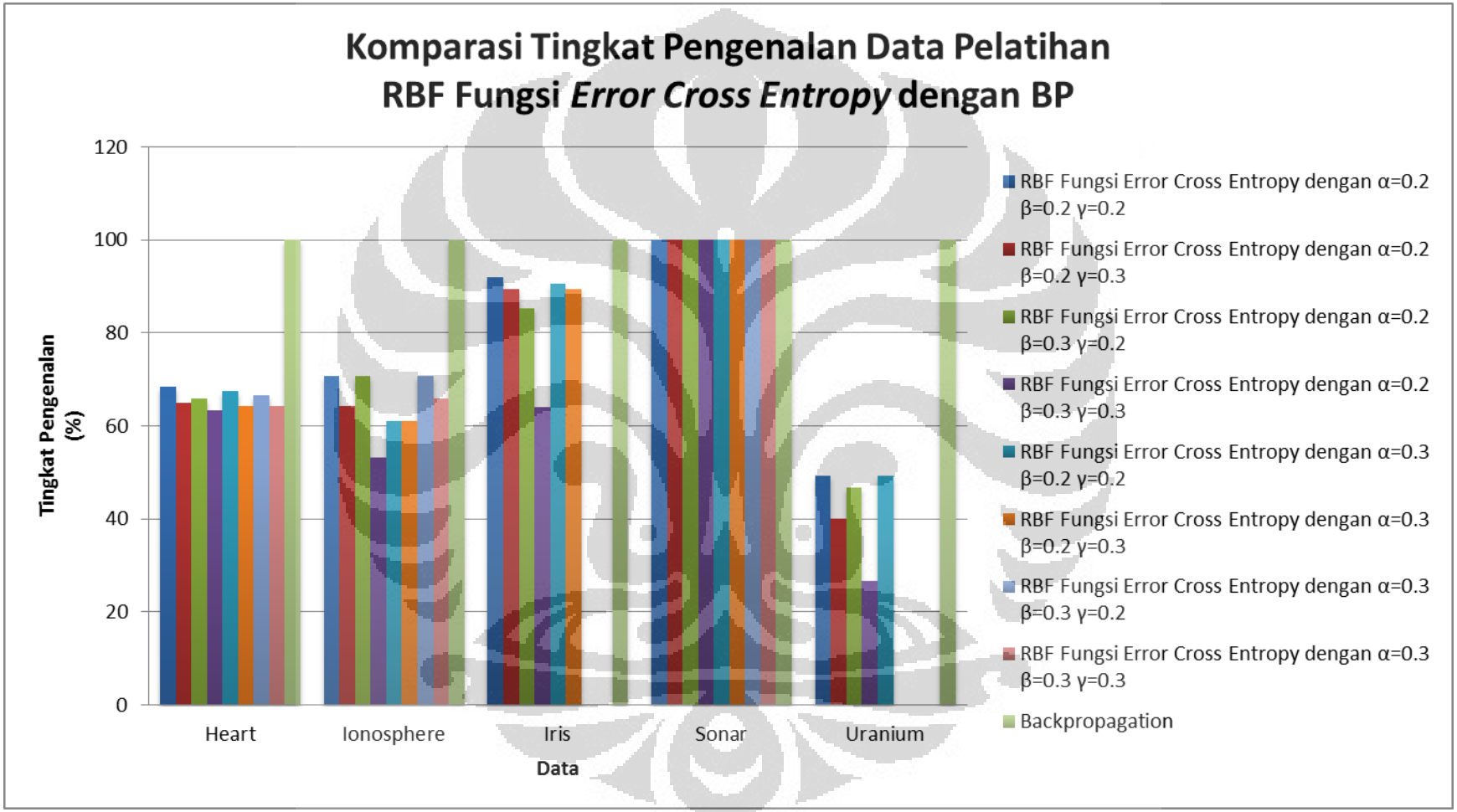
Algoritma *Backpropagation* memiliki jumlah epoh dalam proses pelatihan yang jauh lebih sedikit dibandingkan dengan RBF Fungsi *Error* Kuadratis untuk set data *ionosphere*, *iris*, dan *sonar*. Hal ini terjadi karena proses pelatihan dengan algoritma *Backpropagation* telah mencapai kondisi henti untuk batas *error*, sehingga tidak perlu melakukan pengulangan proses pelatihan sebanyak batas epoh yang telah ditentukan.

Berdasarkan nilai *error* minimum dalam proses pelatihan, algoritma *Backpropagation* memiliki *error* minimum yang lebih baik secara keseluruhan dibandingkan dengan RBF Fungsi *Error* Kuadratis. Hal ini terjadi karena pada

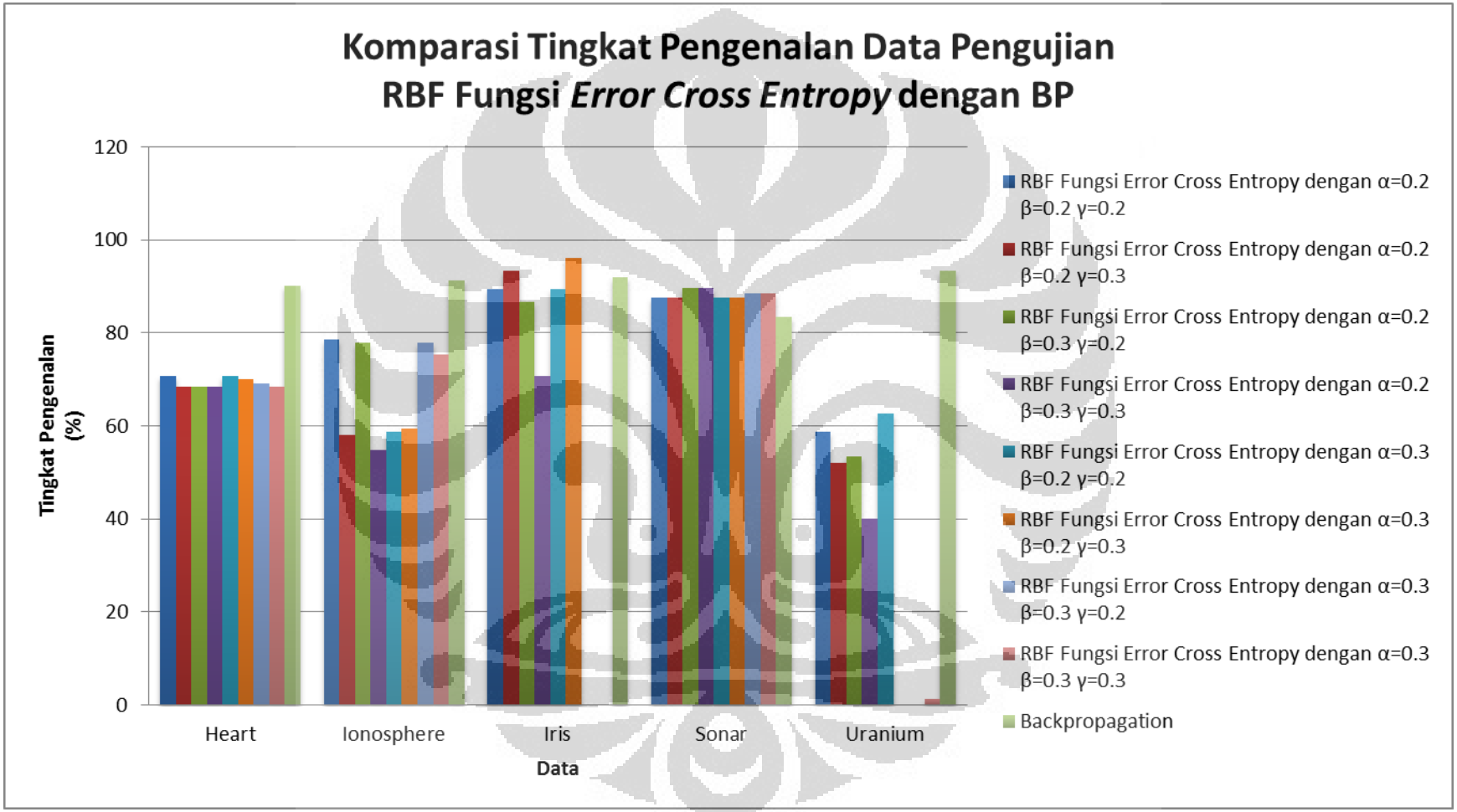
proses penghitungan Δσ di RBF menggunakan satu σ awal yang dibuat ke dalam matriks (1 x jumlah kelas) agar dapat dikoreksi berdasarkan penggunaan di tiap kelas data. Kemudian didapatkan tiga nilai Δσ yang berbeda, namun diambil nilai rata-rata dari ketiga nilai Δσ untuk kemudian di-*update* pada nilai σ awal dan begitu seterusnya. Karena hal tersebutlah nilai *error* minimum yang dicapai tidak sebaik *Backpropagation*, tetapi nilai *error* yang didapat RBF bersifat konvergen setiap epoh. Berbeda jika kita menggunakan nilai Δσ yang tidak mengalami pengambilan nilai rata-rata atau diganti dengan mengambil nilai minimum atau maksimum dari Δσ yang ada. Itu dapat menyebabkan *error* yang dihasilkan tidak bersifat konvergen dan bahkan mengalami osilasi.

5.3 Komparasi Hasil Jaringan Saraf Tiruan dengan Algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy* terhadap Algoritma *Backpropagation*

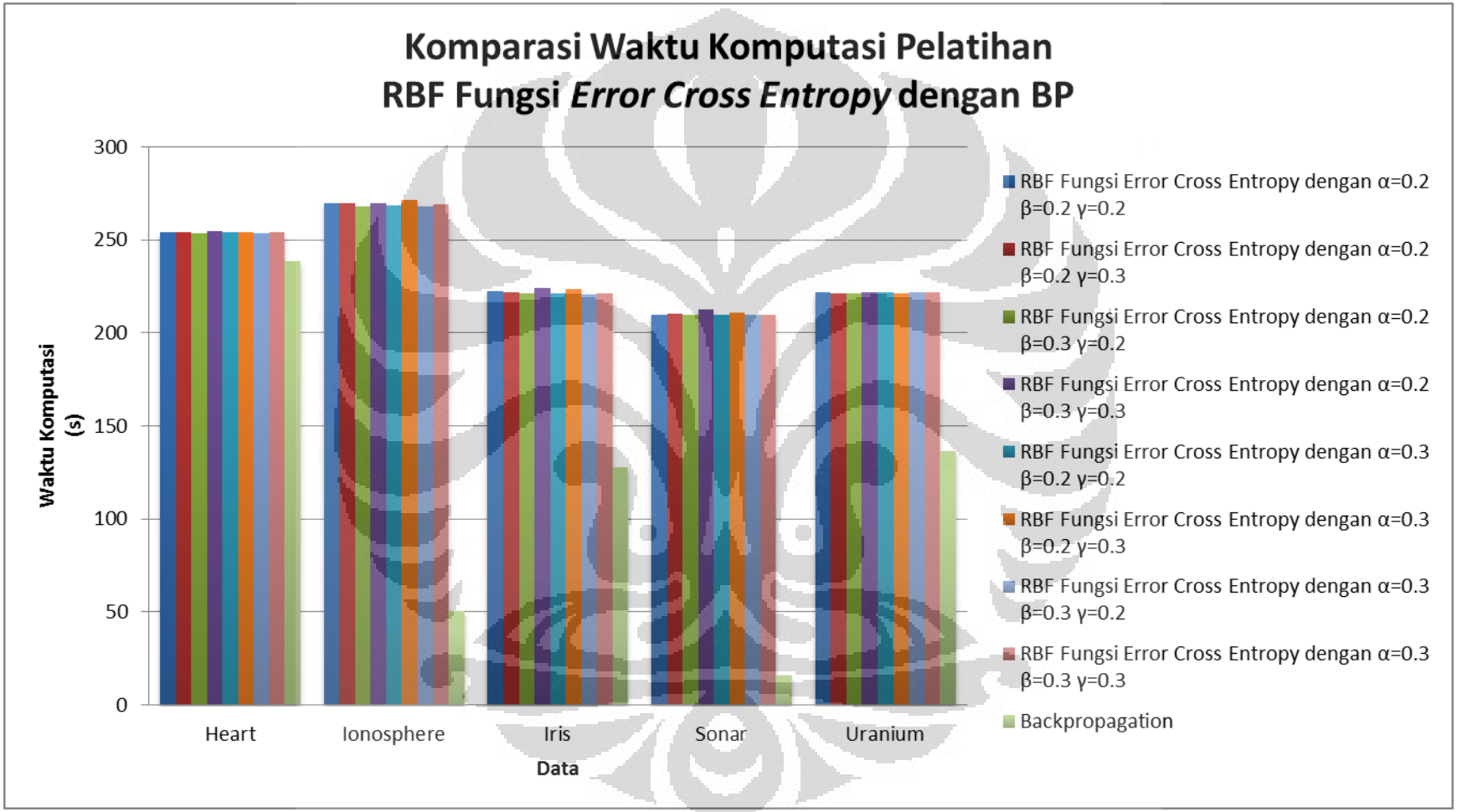
Terdapat batasan pada komparasi ini, yakni data yang dikomparasi adalah data hasil dari JST RBF dengan *alpha* (α), *beta* (β),dan *gamma* (γ) bernilai 0.2 terhadap BP dengan *alpha* (α) bernilai 0.2 serta keduanya memiliki batas epoh maksimum sebesar 10000, batas *error* minimum sebesar 0.01. Data yang digunakan adalah data *heart*, *ionosphere*, *iris*, *sonar*, dan uranium. Komparasi yang dilakukan antara lain komparasi terhadap *error* terkecil, pengenalan data pelatihan, pengenalan data pengujian, serta komparasi epoh maksimum. Setelah itu, hasil komparasi akan dianalisis.



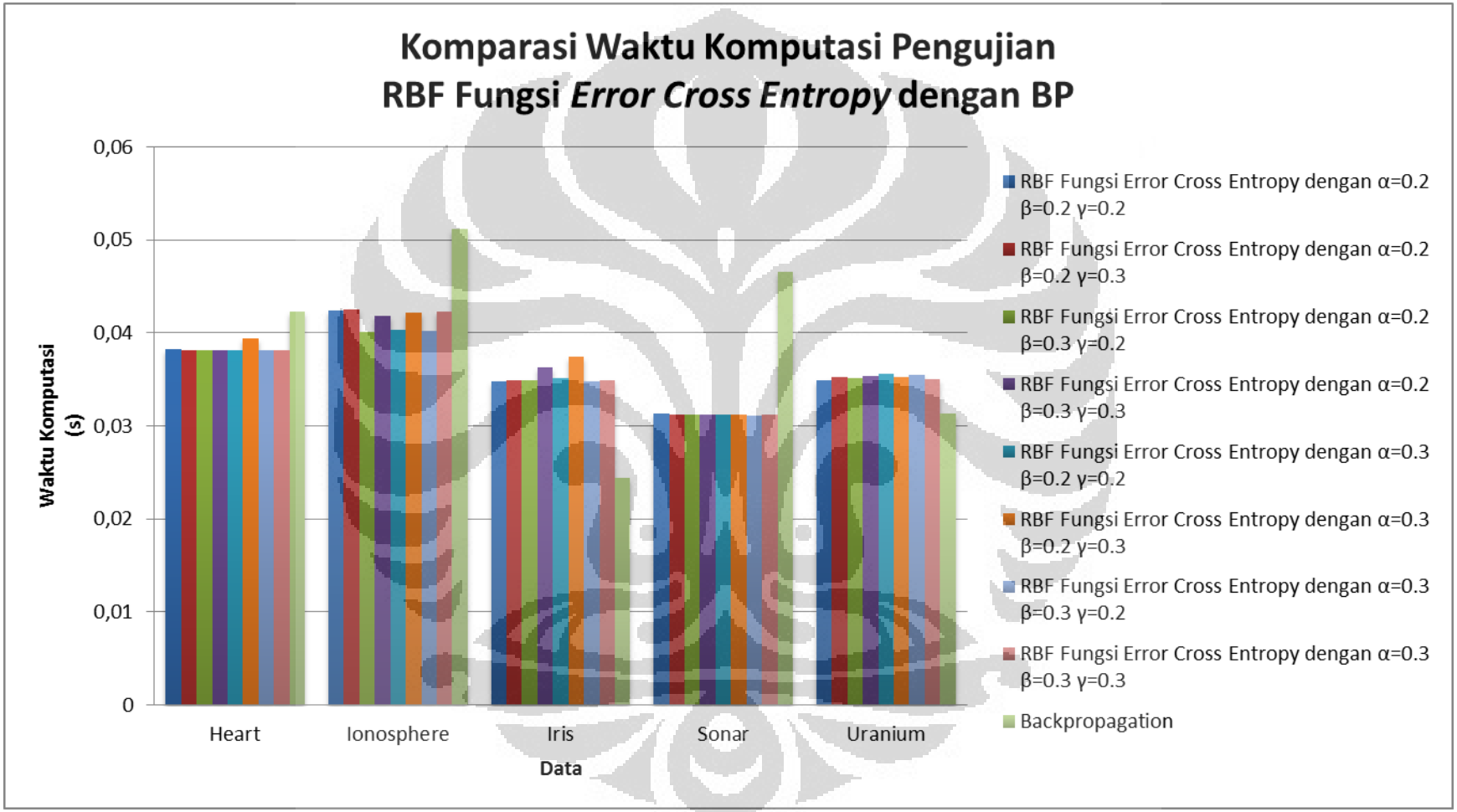
Gambar 5.7 Komparasi Tingkat Pengenalan Data Pelatihan RBF Fungsi *Error Cross-Entropy* dengan BP



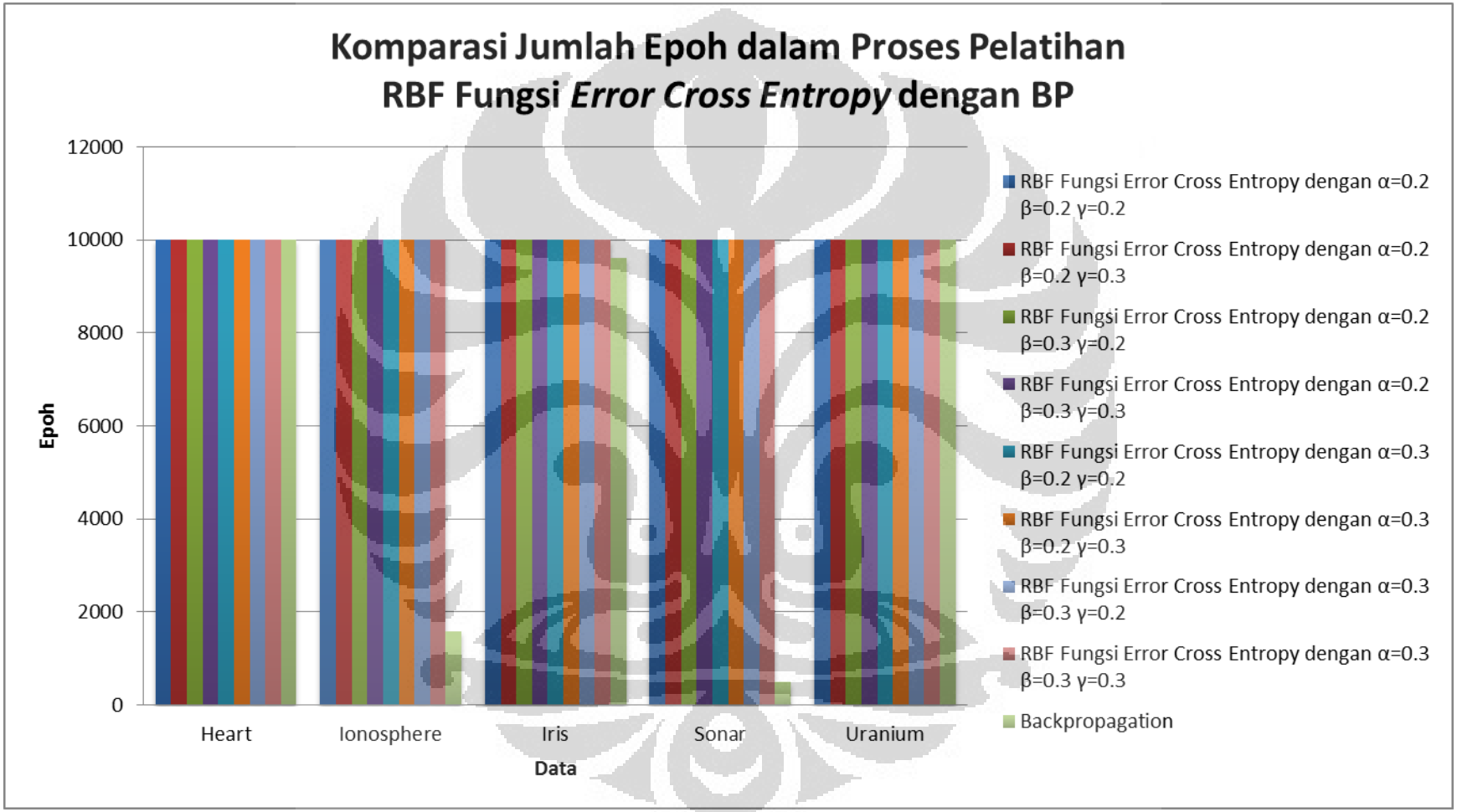
Gambar 5.8 Komparasi Tingkat Pengenalan Data Pengujian RBF Fungsi *Error Cross-Entropy* dengan BP



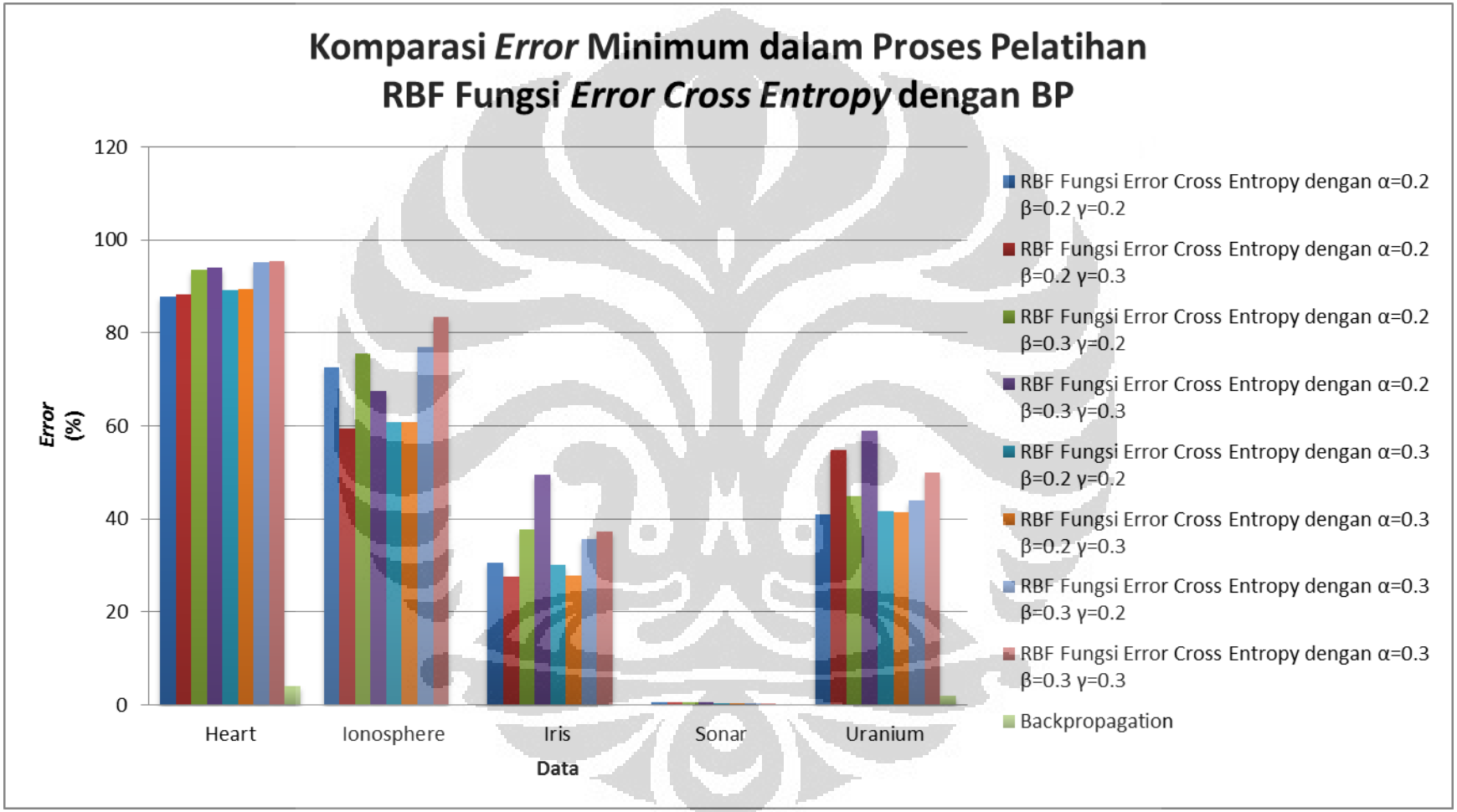
Gambar 5.9 Komparasi Waktu Komputasi Pelatihan RBF Fungsi *Error Cross-Entropy* dengan BP



Gambar 5.10 Komparasi Waktu Komputasi Pengujian RBF Fungsi *Error Cross-Entropy* dengan BP



Gambar 5.11 Komparasi Jumlah Epoh dalam Proses Pelatihan RBF Fungsi *Error Cross-Entropy* dengan BP



Gambar 5.12 Komparasi *Error* Minimum dalam Proses Pelatihan RBF Fungsi *Error Cross-Entropy* dengan BP

5.4 Analisis Komparasi Hasil Jaringan Saraf Tiruan dengan Algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy* terhadap Algoritma *Backpropagation*

Subbab ini akan membahas mengenai komparasi dari hasil RBF fungsi *error cross-entropy* dengan *Backpropagation*. Hasil yang dikomparasi antara lain adalah tingkat pengenalan data pelatihan dan data pengujian, waktu komputasi pelatihan dan pengujian, jumlah epoh dalam proses pelatihan, serta *error* minimum dalam proses pelatihan. Hasil komparasi untuk tingkat pengenalan data pelatihan dan pengujian menunjukkan bahwa hanya set data *iris* dan *sonar* yang dapat mendekati hasil dari algoritma *Backpropagation*. Set data *ionosphere* pun dapat mendekati, akan tetapi perbedaannya (selisih) dapat mencapai 17% (hanya pada percobaan dengan *alpha* 0.2, *beta* 0.2, dan *gamma* 0.2 serta percobaan dengan *alpha* 0.2, *beta* 0.2, dan *gamma* 0.3) untuk data pelatihan dan 10% untuk data pengujian.

Komparasi waktu komputasi pelatihan menunjukkan bahwa *Backpropagation* lebih cepat daripada RBF fungsi *error cross-entropy*. Hal itu terjadi karena pada algoritma *Backpropagation*, proses pelatihan berhenti saat *error*-nya telah mencapai kondisi berhenti. Jika proses pelatihan hanya dibatasi dengan jumlah epoh saja, maka waktu komputasi pada RBF fungsi *error cross-entropy* akan lebih cepat daripada algoritma *Backpropagation*. Hal itu dapat dilihat pada waktu komputasi pengujian, waktu komputasi pengujian untuk set data *heart*, *ionosphere*, dan *sonar* lebih cepat daripada dengan algoritma *Backpropagation*. Namun, pada set data *iris* dan uranium justru terjadi sebaliknya.

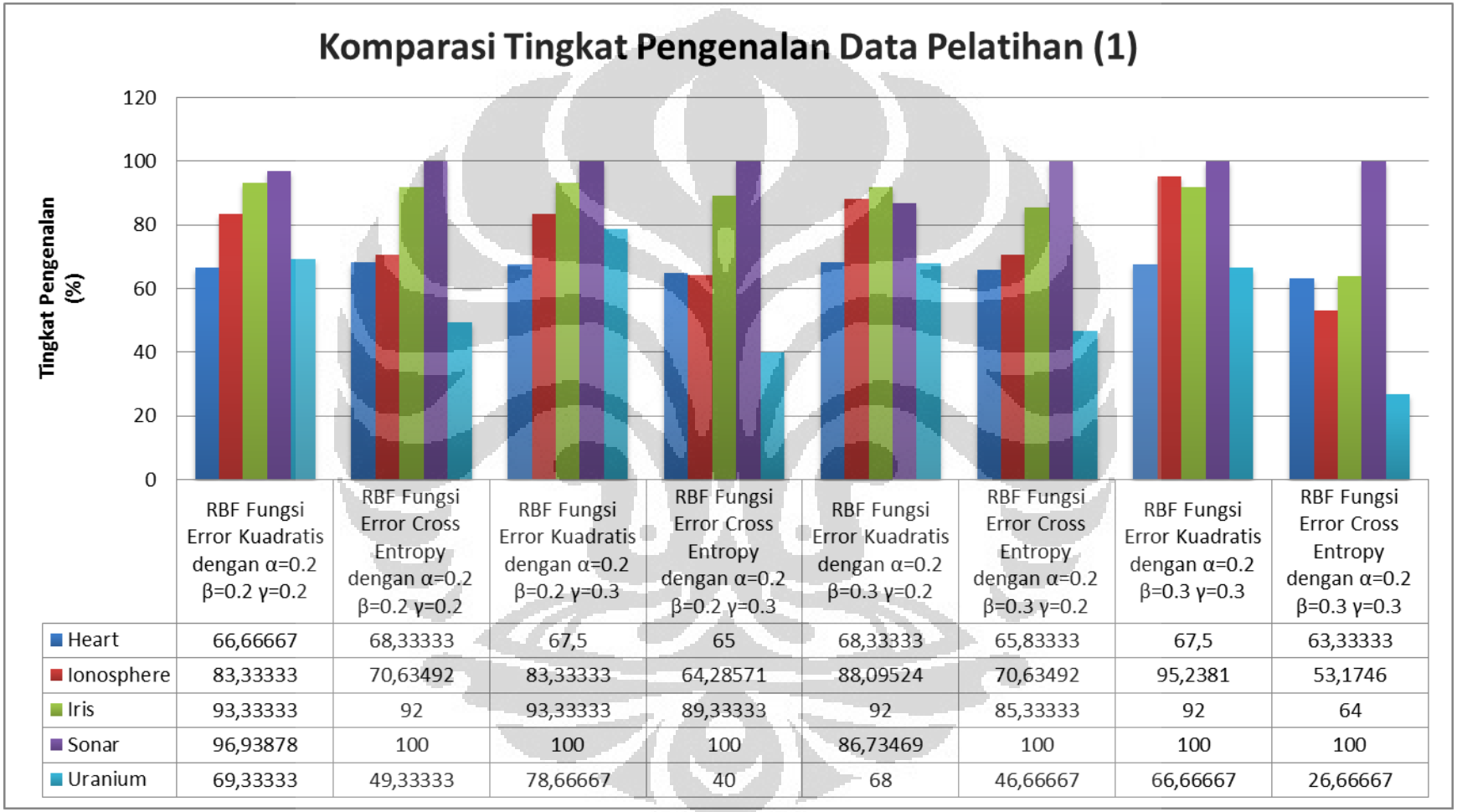
Algoritma *Backpropagation* memiliki jumlah epoh dalam proses pelatihan yang jauh lebih sedikit dibandingkan dengan RBF fungsi *error cross-entropy* untuk set data *ionosphere*, *iris*, dan *sonar*. Hal ini terjadi karena proses pelatihan dengan algoritma *Backpropagation* telah mencapai kondisi henti untuk batas *error*, sehingga tidak perlu melakukan iterasi sebanyak batas epoh yang telah ditentukan.

Berdasarkan besar nilai *error* minimum dalam proses pelatihan, algoritma *Backpropagation* memiliki *error* minimum yang lebih baik secara keseluruhan dibandingkan dengan RBF fungsi *error cross-entropy*. Hal ini terjadi

karena pada proses penghitungan Δσ di RBF menggunakan satu σ awal yang dibuat ke dalam matriks 1 x jumlah kelas agar dapat dikoreksi berdasarkan penggunaan di tiap kelas data. Kemudian didapatkan tiga nilai Δσ yang berbeda. Nilai-nilai Δσ tersebut kemudian dirata-rata untuk kemudian di-*update* pada nilai σ awal dan begitu seterusnya. Karena hal tersebutlah nilai *error* minimum yang dicapai tidak sebaik *Backpropagation*, nilai *error* yang didapat RBF bersifat konvergen setiap epoh, berbeda jika kita menggunakan nilai Δσ yang tidak mengalami pengambilan nilai rata-rata atau diganti dengan mengambil nilai minimum atau maksimum dari Δσ yang ada. Hal tersebut dapat menyebabkan *error* yang dihasilkan tidak bersifat konvergen dan bahkan mengalami osilasi.

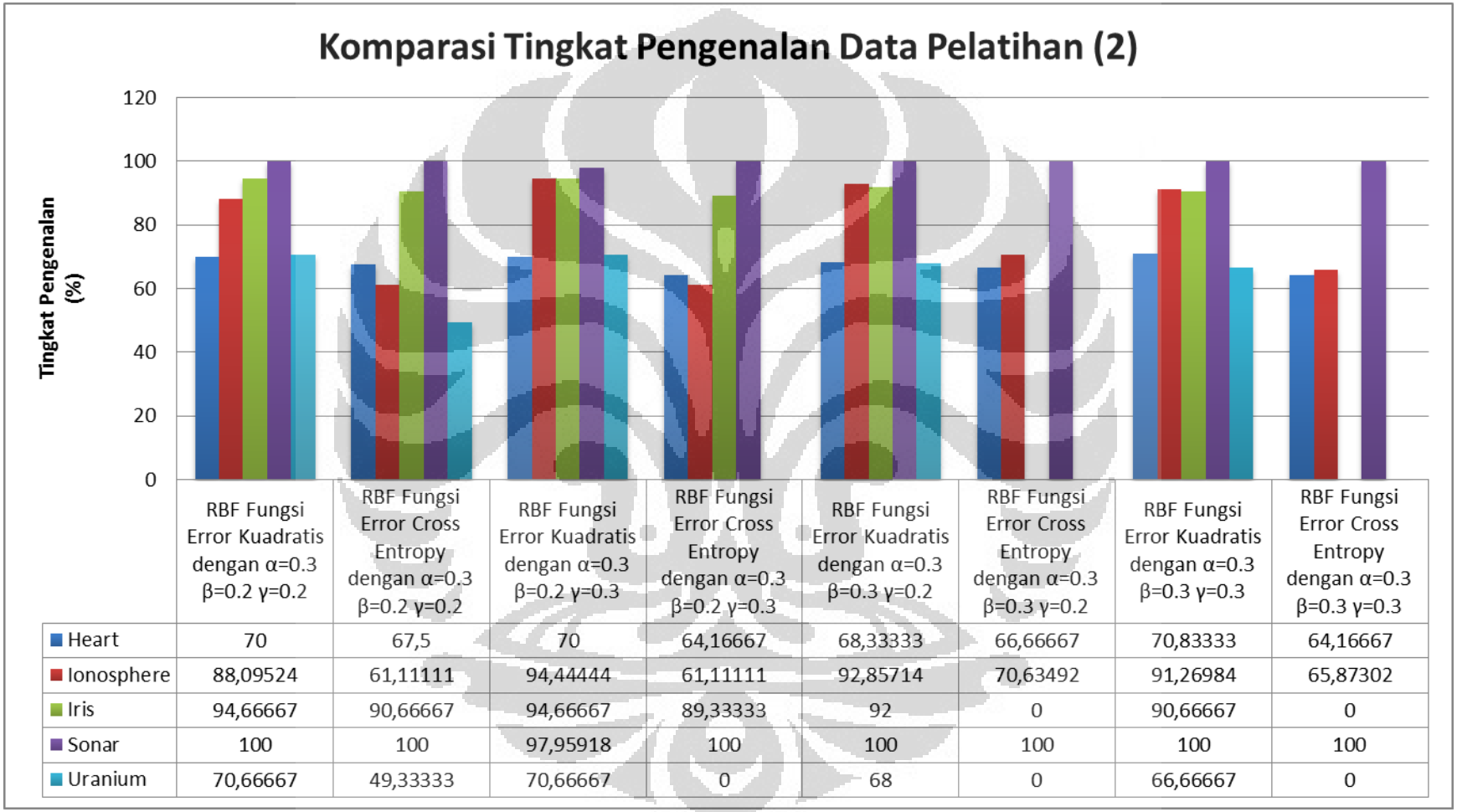
5.5 Komparasi Hasil Jaringan Saraf Tiruan dengan Algoritma *Radial Basis Function* Fungsi *Error* Kuadratis terhadap Algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy*

Terdapat batasan pada komparasi ini, yakni data yang dikomparasi adalah data hasil dari JST RBF fungsi *error* kuadratis dengan JST RBF fungsi *error cross-entropy* dengan parameter-parameter yang sama seperti *alpha* sebesar 0.2 untuk ketiganya (untuk RBF *beta* dan *gamma* bernilai 0.2), batas epoh maksimum sebesar 10000, batas *error* minimum sebesar 0.01. Data yang digunakan adalah data *heart*, *ionosphere*, *iris*, *sonar*, dan uranium. Komparasi yang dilakukan antara lain adalah komparasi terhadap *error* terkecil, pengenalan data pelatihan, pengenalan data pengujian, serta komparasi epoh maksimum.



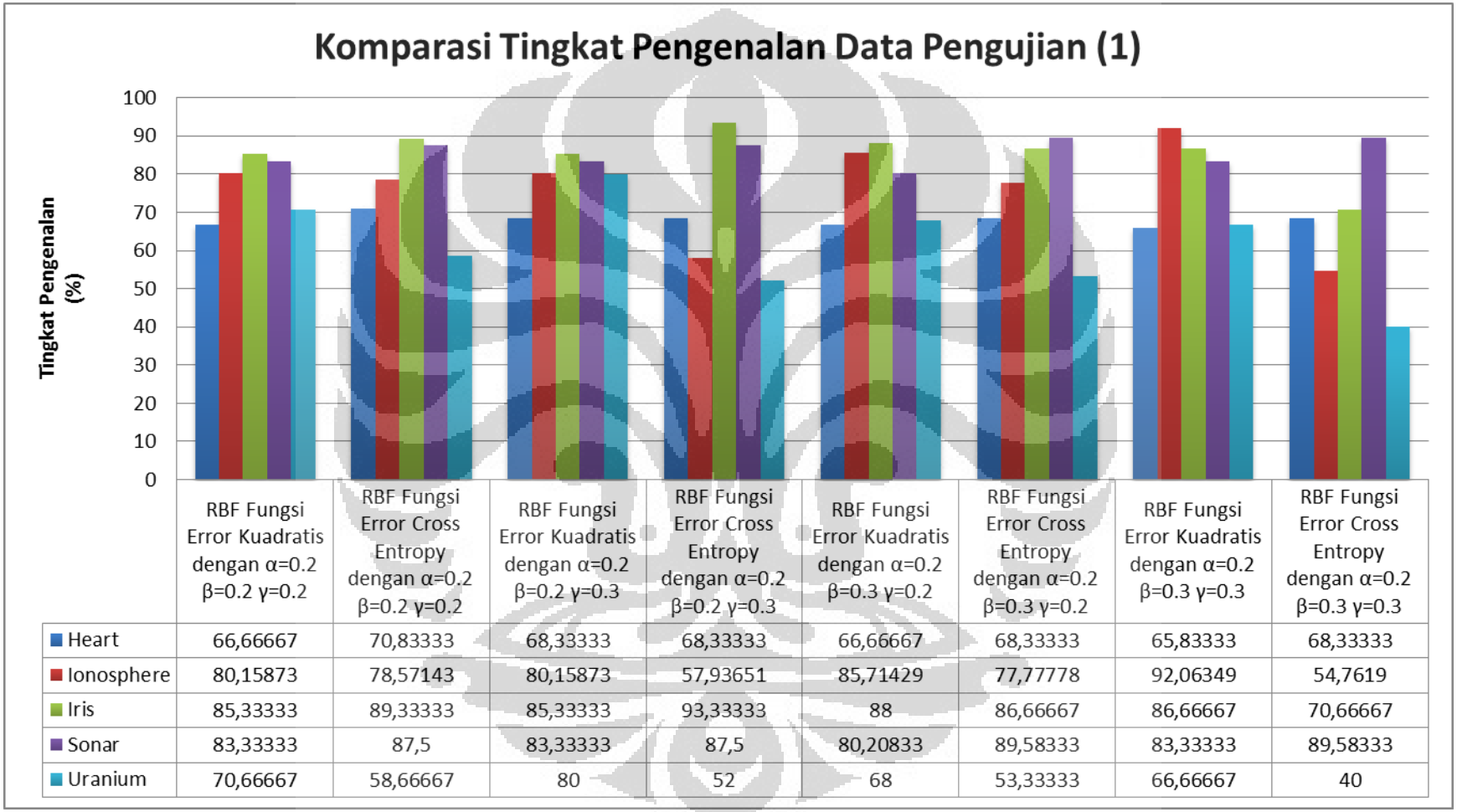
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	66,66667	68,33333	67,5	65	68,33333	65,83333	67,5	63,33333
Ionosphere	83,33333	70,63492	83,33333	64,28571	88,09524	70,63492	95,2381	53,1746
Iris	93,33333	92	93,33333	89,33333	92	85,33333	92	64
Sonar	96,93878	100	100	100	86,73469	100	100	100
Uranium	69,33333	49,33333	78,66667	40	68	46,66667	66,66667	26,66667

Gambar 5.13 Komparasi Tingkat Pengenalan Data Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)



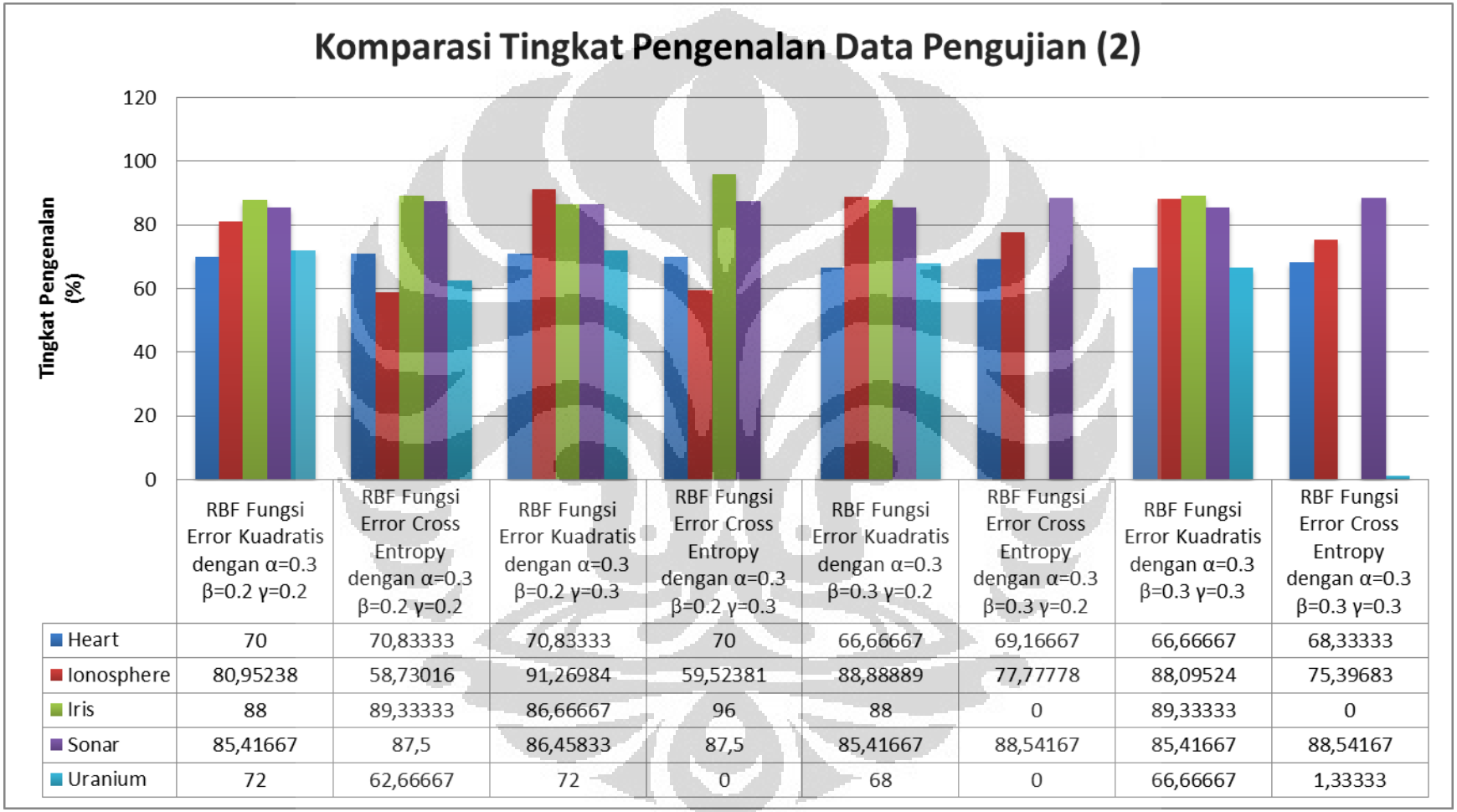
	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	70	67,5	70	64,16667	68,33333	66,66667	70,83333	64,16667
Ionosphere	88,09524	61,11111	94,44444	61,11111	92,85714	70,63492	91,26984	65,87302
Iris	94,66667	90,66667	94,66667	89,33333	92	0	90,66667	0
Sonar	100	100	97,95918	100	100	100	100	100
Uranium	70,66667	49,33333	70,66667	0	68	0	66,66667	0

Gambar 5.14 Komparasi Tingkat Pengenalan Data Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)



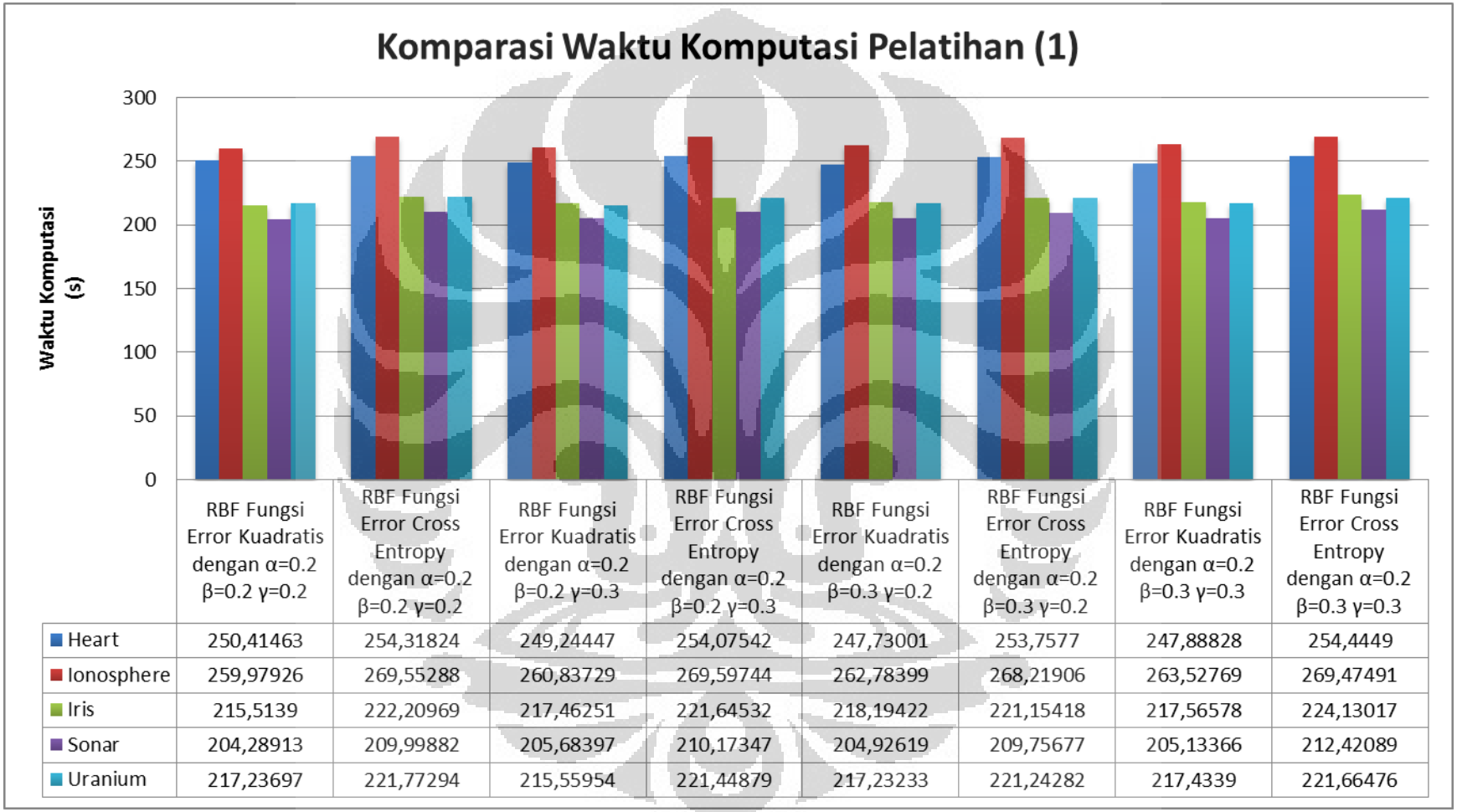
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	66,66667	70,83333	68,33333	68,33333	66,66667	68,33333	65,83333	68,33333
Ionosphere	80,15873	78,57143	80,15873	57,93651	85,71429	77,77778	92,06349	54,7619
Iris	85,33333	89,33333	85,33333	93,33333	88	86,66667	86,66667	70,66667
Sonar	83,33333	87,5	83,33333	87,5	80,20833	89,58333	83,33333	89,58333
Uranium	70,66667	58,66667	80	52	68	53,33333	66,66667	40

Gambar 5.15 Komparasi Tingkat Pengenalan Data Pengujian RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)



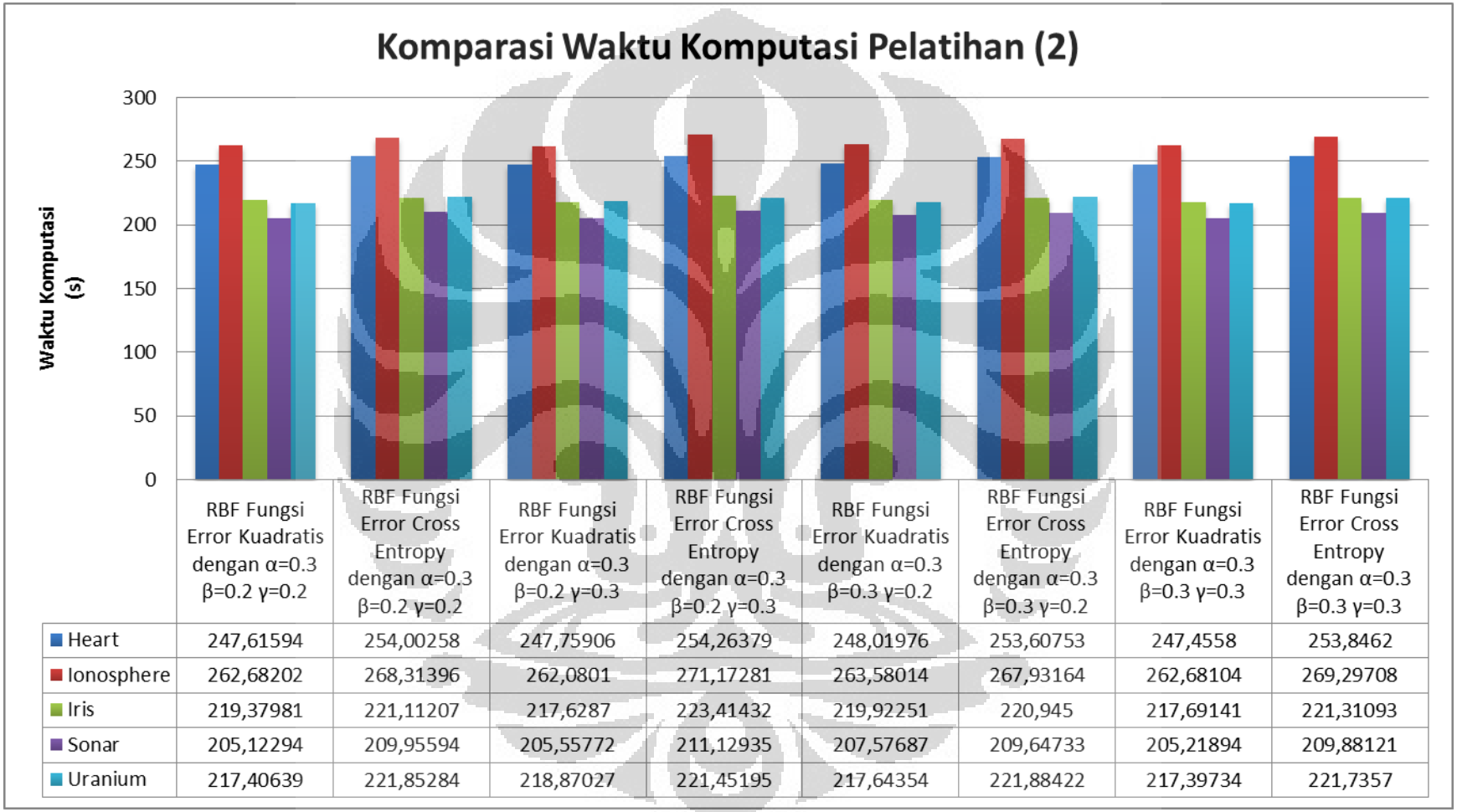
	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	70	70,83333	70,83333	70	66,66667	69,16667	66,66667	68,33333
Ionosphere	80,95238	58,73016	91,26984	59,52381	88,88889	77,77778	88,09524	75,39683
Iris	88	89,33333	86,66667	96	88	0	89,33333	0
Sonar	85,41667	87,5	86,45833	87,5	85,41667	88,54167	85,41667	88,54167
Uranium	72	62,66667	72	0	68	0	66,66667	1,33333

Gambar 5.16 Komparasi Tingkat Pengenalan Data Pengujian RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)



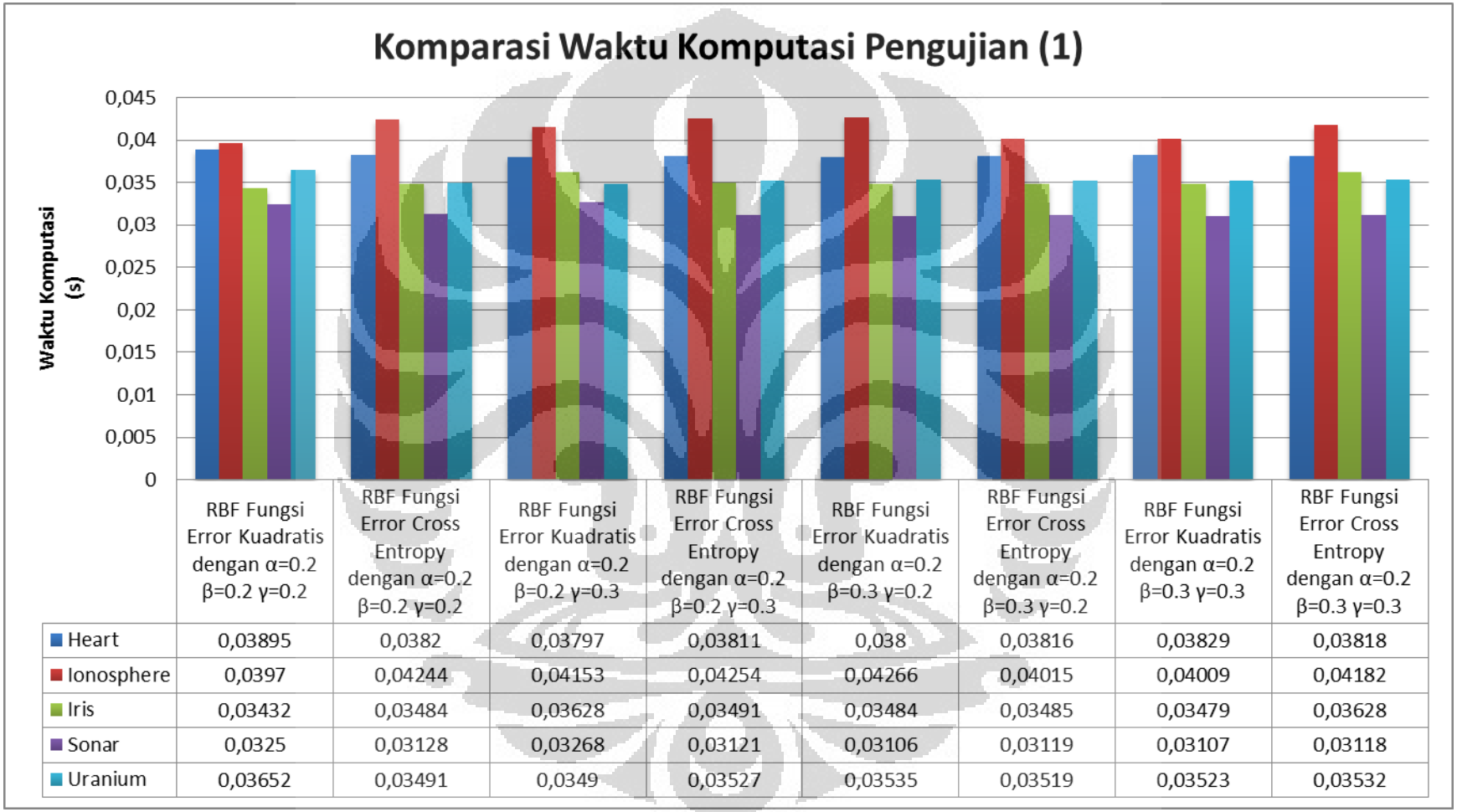
	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	250,41463	254,31824	249,24447	254,07542	247,73001	253,7577	247,88828	254,4449
Ionosphere	259,97926	269,55288	260,83729	269,59744	262,78399	268,21906	263,52769	269,47491
Iris	215,5139	222,20969	217,46251	221,64532	218,19422	221,15418	217,56578	224,13017
Sonar	204,28913	209,99882	205,68397	210,17347	204,92619	209,75677	205,13366	212,42089
Uranium	217,23697	221,77294	215,55954	221,44879	217,23233	221,24282	217,4339	221,66476

Gambar 5.17 Komparasi Waktu Komputasi Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)



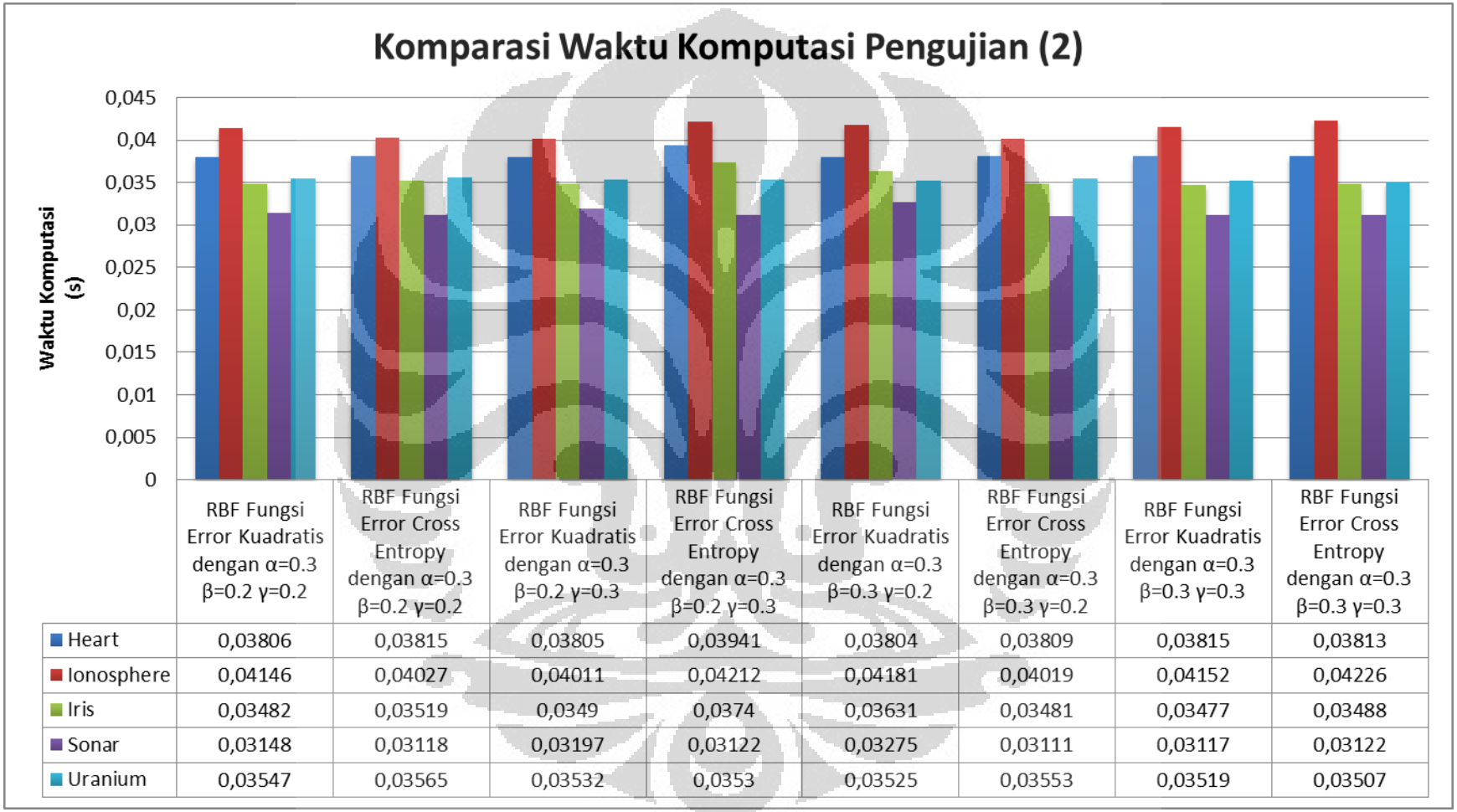
	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	247,61594	254,00258	247,75906	254,26379	248,01976	253,60753	247,4558	253,8462
Ionosphere	262,68202	268,31396	262,0801	271,17281	263,58014	267,93164	262,68104	269,29708
Iris	219,37981	221,11207	217,6287	223,41432	219,92251	220,945	217,69141	221,31093
Sonar	205,12294	209,95594	205,55772	211,12935	207,57687	209,64733	205,21894	209,88121
Uranium	217,40639	221,85284	218,87027	221,45195	217,64354	221,88422	217,39734	221,7357

Gambar 5.18 Komparasi Waktu Komputasi Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)



	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	0,03895	0,0382	0,03797	0,03811	0,038	0,03816	0,03829	0,03818
Ionosphere	0,0397	0,04244	0,04153	0,04254	0,04266	0,04015	0,04009	0,04182
Iris	0,03432	0,03484	0,03628	0,03491	0,03484	0,03485	0,03479	0,03628
Sonar	0,0325	0,03128	0,03268	0,03121	0,03106	0,03119	0,03107	0,03118
Uranium	0,03652	0,03491	0,0349	0,03527	0,03535	0,03519	0,03523	0,03532

Gambar 5.19 Komparasi Waktu Komputasi Pengujian RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)



	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	0,03806	0,03815	0,03805	0,03941	0,03804	0,03809	0,03815	0,03813
Ionosphere	0,04146	0,04027	0,04011	0,04212	0,04181	0,04019	0,04152	0,04226
Iris	0,03482	0,03519	0,0349	0,0374	0,03631	0,03481	0,03477	0,03488
Sonar	0,03148	0,03118	0,03197	0,03122	0,03275	0,03111	0,03117	0,03122
Uranium	0,03547	0,03565	0,03532	0,0353	0,03525	0,03553	0,03519	0,03507

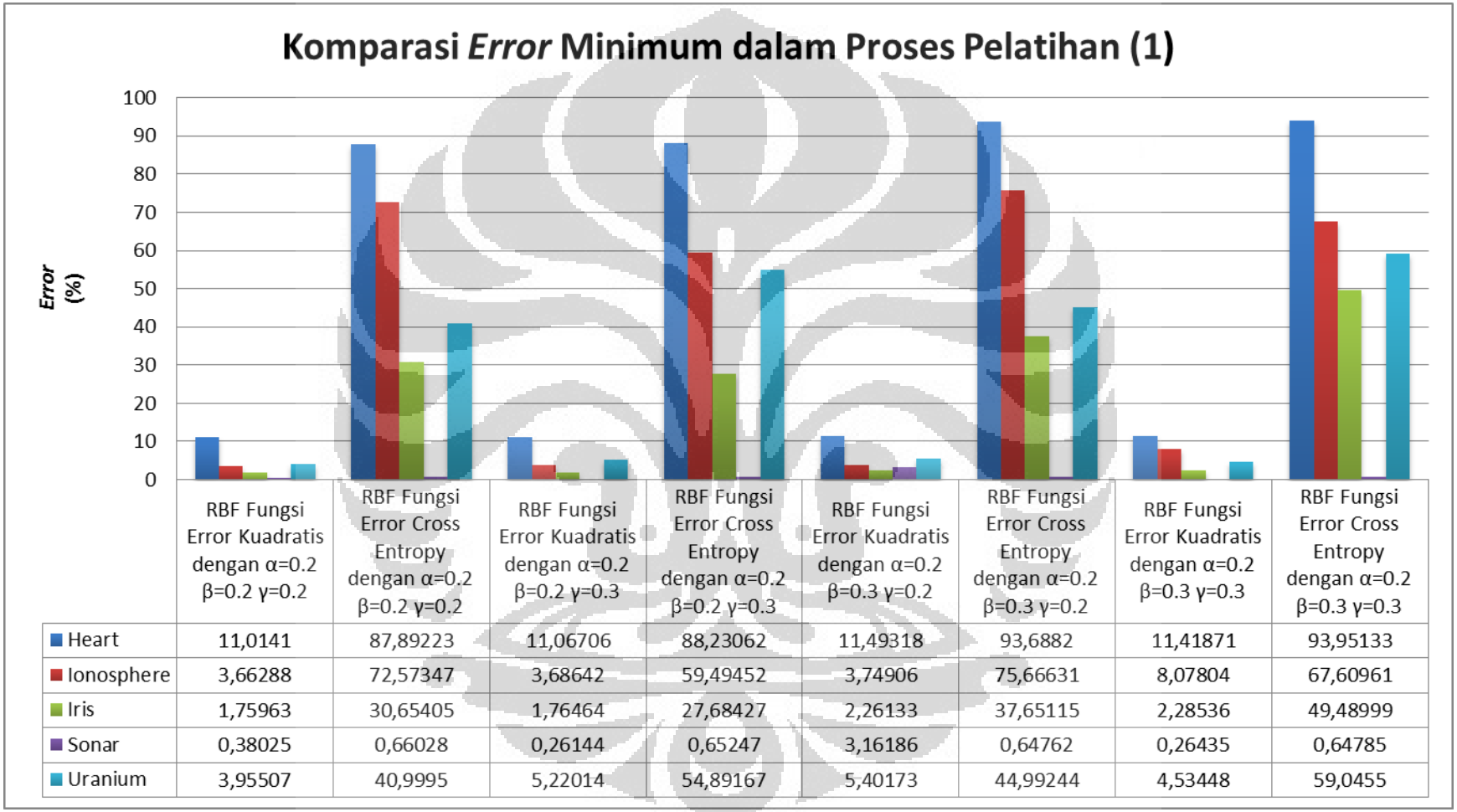
Gambar 5.20 Komparasi Waktu Komputasi Pengujian RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)

	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	10000	10000	10000	10000	10000	10000	10000	10000
Ionosphere	10000	10000	10000	10000	10000	10000	10000	10000
Iris	10000	10000	10000	10000	10000	10000	10000	10000
Sonar	10000	10000	10000	10000	10000	10000	10000	10000
Uranium	10000	10000	10000	10000	10000	10000	10000	10000

Gambar 5.21 Komparasi Jumlah Epoh dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)

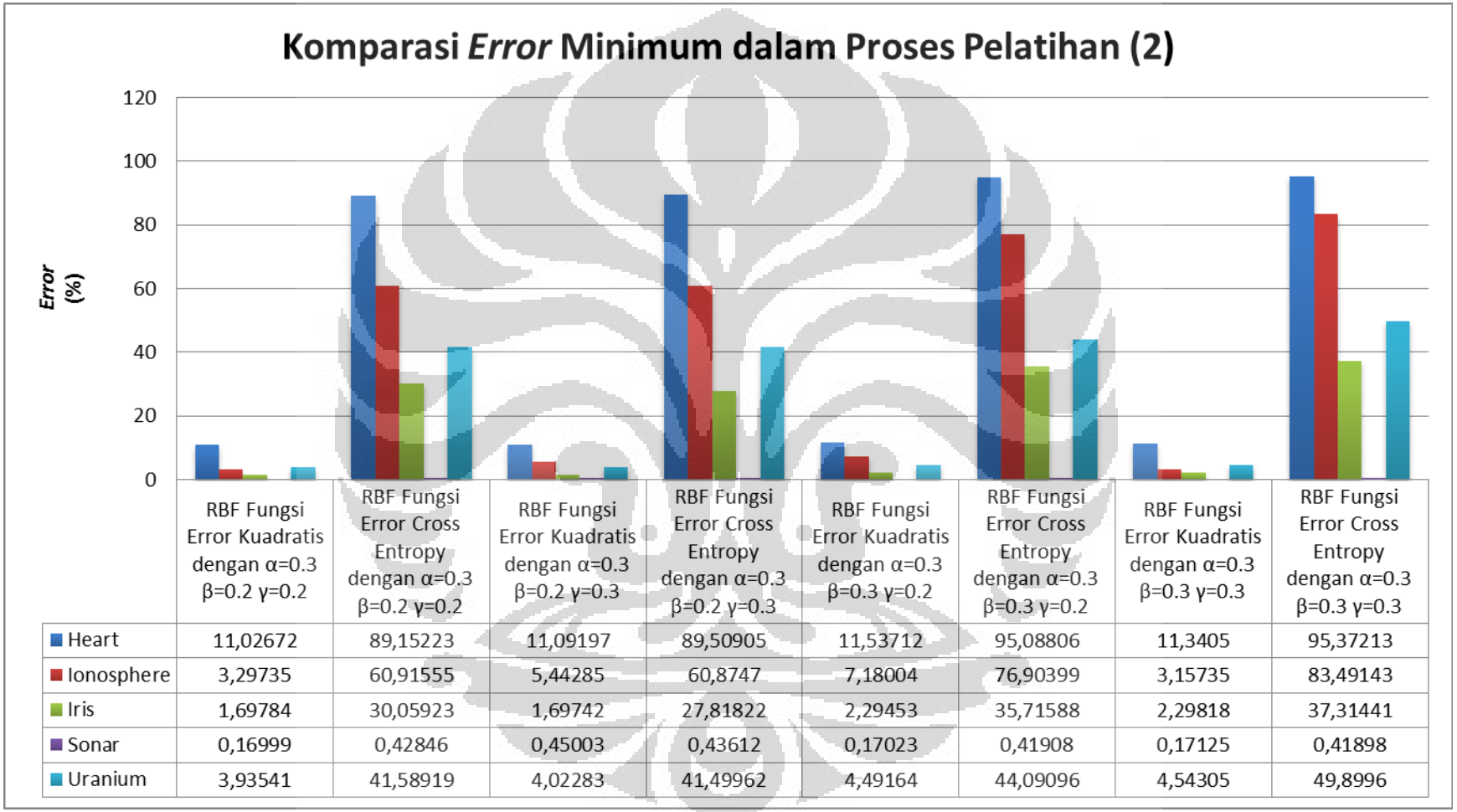
	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	10000	10000	10000	10000	10000	10000	10000	10000
Ionosphere	10000	10000	10000	10000	10000	10000	10000	10000
Iris	10000	10000	10000	10000	10000	10000	10000	10000
Sonar	10000	10000	10000	10000	10000	10000	10000	10000
Uranium	10000	10000	10000	10000	10000	10000	10000	10000

Gambar 5.22 Komparasi Jumlah Epoh dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)



	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.2 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.2 β=0.3 γ=0.3
Heart	11,0141	87,89223	11,06706	88,23062	11,49318	93,6882	11,41871	93,95133
Ionosphere	3,66288	72,57347	3,68642	59,49452	3,74906	75,66631	8,07804	67,60961
Iris	1,75963	30,65405	1,76464	27,68427	2,26133	37,65115	2,28536	49,48999
Sonar	0,38025	0,66028	0,26144	0,65247	3,16186	0,64762	0,26435	0,64785
Uranium	3,95507	40,9995	5,22014	54,89167	5,40173	44,99244	4,53448	59,0455

Gambar 5.23 Komparasi *Error* Minimum dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (1)



	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.2 γ=0.3	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.2	RBF Fungsi Error Kuadratis dengan α=0.3 β=0.3 γ=0.3	RBF Fungsi Error Cross Entropy dengan α=0.3 β=0.3 γ=0.3
Heart	11,02672	89,15223	11,09197	89,50905	11,53712	95,08806	11,3405	95,37213
Ionosphere	3,29735	60,91555	5,44285	60,8747	7,18004	76,90399	3,15735	83,49143
Iris	1,69784	30,05923	1,69742	27,81822	2,29453	35,71588	2,29818	37,31441
Sonar	0,16999	0,42846	0,45003	0,43612	0,17023	0,41908	0,17125	0,41898
Uranium	3,93541	41,58919	4,02283	41,49962	4,49164	44,09096	4,54305	49,8996

Gambar 5.24 Komparasi *Error* Minimum dalam Proses Pelatihan RBF Fungsi *Error* Kuadratis dengan RBF Fungsi *Error Cross-Entropy* (2)

5.6 Analisis Komparasi Hasil Jaringan Saraf Tiruan dengan Algoritma *Radial Basis Function* Fungsi *Error* Kuadratis terhadap Algoritma *Radial Basis Function* Fungsi *Error Cross-Entropy*

Proses komparasi pada percobaan ini dilakukan antar RBF dengan menggunakan parameter-parameter yang bersesuaian, misalnya RBF fungsi *error* kuadratis dengan $\alpha = 0.2$, $\beta = 0.2$ dan $\gamma = 0.2$ terhadap RBF fungsi *error cross-entropy* dengan $\alpha = 0.2$, $\beta = 0.2$ dan $\gamma = 0.2$. Hal ini dilakukan agar didapat hasil komparasi yang bersesuaian. Hasil yang didapat menunjukkan bahwa terdapat perbedaan dalam tingkat pengenalan. RBF dengan fungsi *error cross-entropy* memiliki tingkat pengenalan yang lebih rendah dibandingkan dengan RBF fungsi *error* kuadratis. Hal ini terjadi karena fungsi *error cross-entropy* menghitung besar *error* sesuai dengan persamaan logaritma yang akan menghasilkan *error* yang lebih besar daripada menggunakan fungsi *error* kuadratis, sehingga mengakibatkan tingkat pengenalan yang kurang baik.

Faktor lain yang mempengaruhi tingkat pengenalan set data pelatihan adalah pemilihan fungsi *error*. Fungsi *error* memainkan peranan penting dalam proses pelatihan dan pemakaian fungsi *error* yang tepat akan menghasilkan performa pengenalan data pelatihan yang maksimal. Dengan demikian, apabila data yang digunakan untuk pelatihan cukup mewakili data yang diujikan, maka tingkat pengenalan data pengujian akan memiliki hasil yang sejalan dengan tingkat pengenalan data pelatihan. Hal ini dapat dilihat pada tingkat pengenalan data pelatihan dan pengujian untuk set data *heart* dan *sonar* yang menggunakan RBF fungsi *error cross-entropy* lebih baik daripada hasil dari RBF fungsi *error* kuadratis.

Untuk waktu komputasi baik pada pelatihan ataupun pengujian, RBF fungsi *error cross-entropy* memiliki waktu komputasi yang lebih lama dikarenakan persamaan yang digunakan untuk menghitung *error*-nya lebih sulit dan terdapat momentum sehingga menambah lagi proses aritmatika yang dilakukan.

Jumlah epoh dalam proses pelatihan yang digunakan baik oleh RBF fungsi *error cross-entropy* atau RBF fungsi *error* kuadratis sama-sama sebanyak 10000 epoh. Ini dikarenakan 10000 epoh merupakan batas yang digunakan untuk

melakukan iterasi pada proses pelatihan serta belum tercapainya kondisi berhenti lainnya berupa *error* sebesar 0.01 sebelum epoh mencapai 10000.

Error minimum yang dicapai pada RBF fungsi *error* kuadratis bernilai lebih rendah daripada *error* minimum pada RBF fungsi *error cross-entropy* . Hal ini terjadi karena perhitungan yang digunakan pada RBF fungsi *error cross-entropy* menghasilkan nilai yang lebih besar. Penjelasan tersebut menunjukkan bahwa fungsi *error* sangat berpengaruh terhadap *error* minimum yang didapat pada suatu jaringan saraf tiruan. Selain itu, data pelatihan juga mempengaruhi besarnya *error* minimum karena banyaknya kelas pada suatu set data akan mempengaruhi jumlah neuron pada lapisan tersembunyi yang digunakan untuk menghitung jarak antara data dengan nilai tengah dan lebar data tiap kelas. Berdasarkan hal tersebut, jelas sekali bahwa proses aritmatika dipengaruhi oleh jumlah neuron pada lapisan tersembunyi yang berhubungan dengan set data yang digunakan. Semakin kecil *error* minimum yang diperoleh, maka semakin baik performa dari suatu jaringan.

KESIMPULAN

Berdasarkan percobaan dan analisis yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pemilihan fungsi *error* yang digunakan berpengaruh pada performa pada jaringan saraf tiruan *Radial Basis Function*.
2. Salah satu faktor yang mempengaruhi tingkat pengenalan data pada jaringan saraf tiruan *Radial Basis Function*, adalah pemilihan fungsi *error.*
3. Waktu komputasi dipengaruhi oleh beberapa faktor, yaitu:
    a. Jumlah data yang digunakan
    b. Jumlah atribut data yang digunakan
    c. Jumlah kelas set data yang digunakan
    d. Pemilihan fungsi er*rror*
    e. Jumlah epoh dalam proses pelatihan
4. Jaringan Saraf Tiruan *Radial Basis Function* memiliki waktu komputasi yang lebih cepat dibandingkan dengan Jaringan Saraf Tiruan *Backpropagation* yang dapat dilihat pada waktu komputasi pengujian dan waktu komputasi untuk setiap epoh yang dilakukan atau jika proses pelatihan dibatasi dengan jumlah epoh saja.

DAFTAR REFERENSI

[1] Adityamurthi, D. (2011). *Pengembangan Algortima Radial Basis Function dengan Fungsi Error Cross-Entrpoy pada Jaringan Saraf Tiruan Tunggal dan Ensemble serta Perbandingannya dengan Backpropagation.* Depok: Universitas Indonesia.

[2] Bors, A. G. (n.d.). *Introduction of the Radial Basis Function (RBF) Networks*. York, UK.

[3] Fu, L. M. (1994). *Neural Networks in Computer Intelligence.* Singapore: McGraw-Hill.

[4] *Radial Basis Function* (RBF) *Networks*. (n.d.). http://ccy.dd.ncu.edu.tw/~chen/course/Neural/ch5/RBF.htm

[5] Rafflesia, U. (2010, Juli 27). Perbandingan Performansi Jaringan *Learning Vector Quantization* (LVQ) dan *Radial Basis Function* (RBF) untuk Permasalahan Klasifikasi Penyakit Karies Gigi. Surabaya. http://www.scribd.com/doc/88718747/ITS-Master-14681-1208201012-Presentation

[6] Sarwono, Y. T. (n.d.). Aplikasi Model Jaringan Saraf Tiruan dengan *Radial Basis Function* untuk Mendeteksi Kelainan Otak (*Stroke Infark*).

[7] Sjoberg, J. (2005). *Mathematica Neural Netwroks.* Wolfram Research, Inc.