**LAPORAN PRAKTIKUM DATA WAREHOUSING DAN DATA**

**MINING**

**PERTEMUAN 7**

**“ALGORITMA KLASIFIKASI : ARTIFICAL NEURAL NETWORK”**



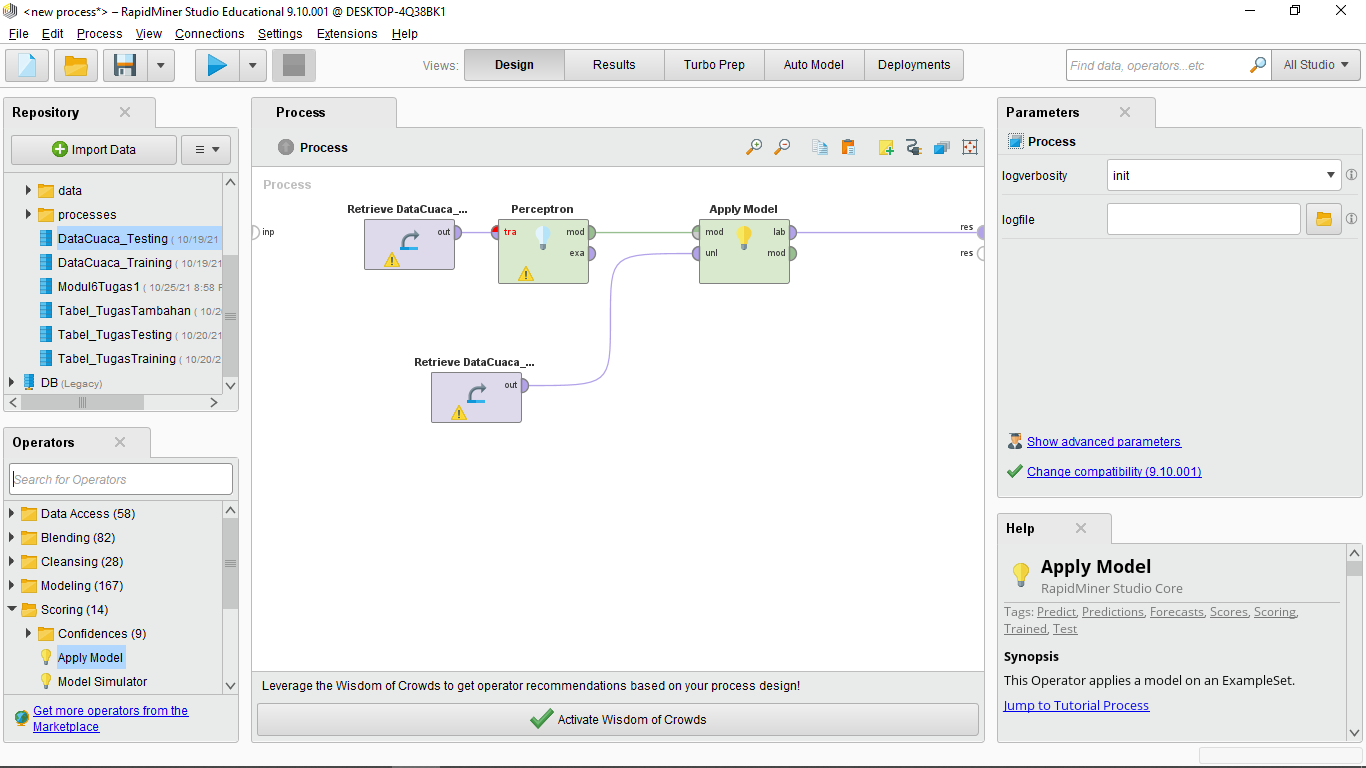
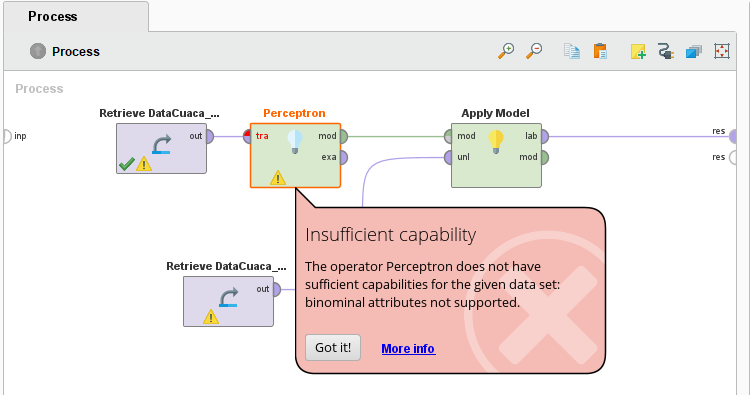
**Oleh:**

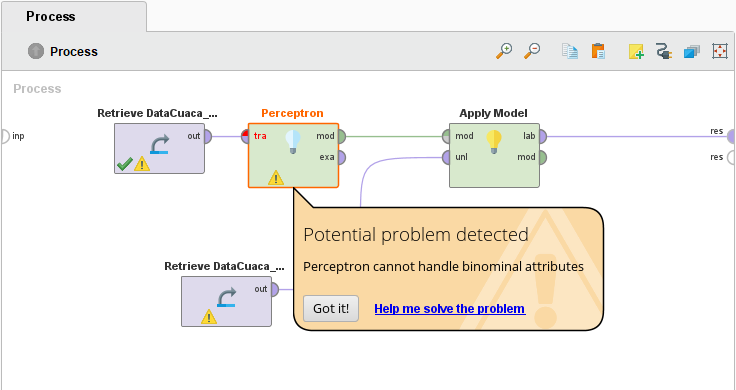
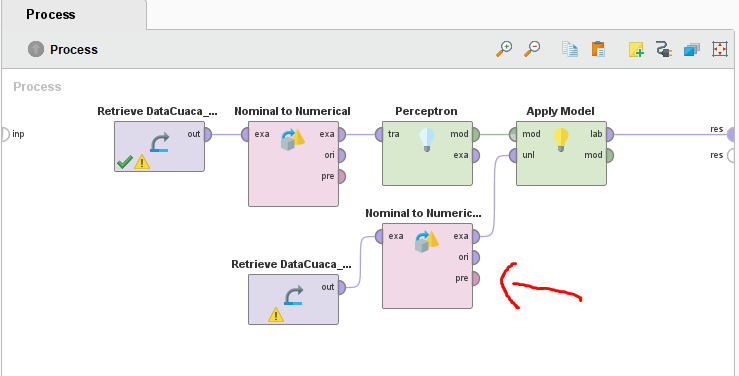
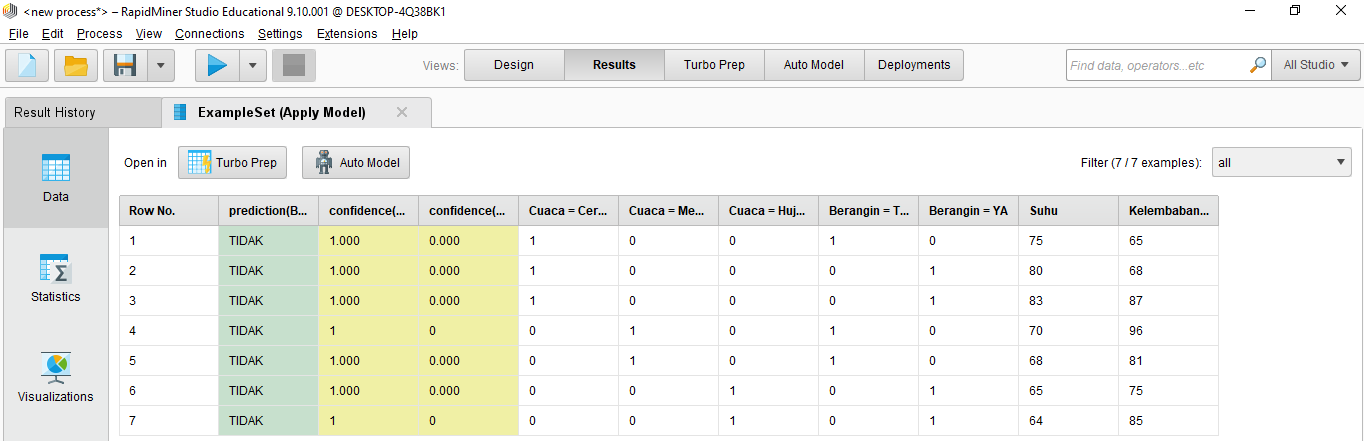
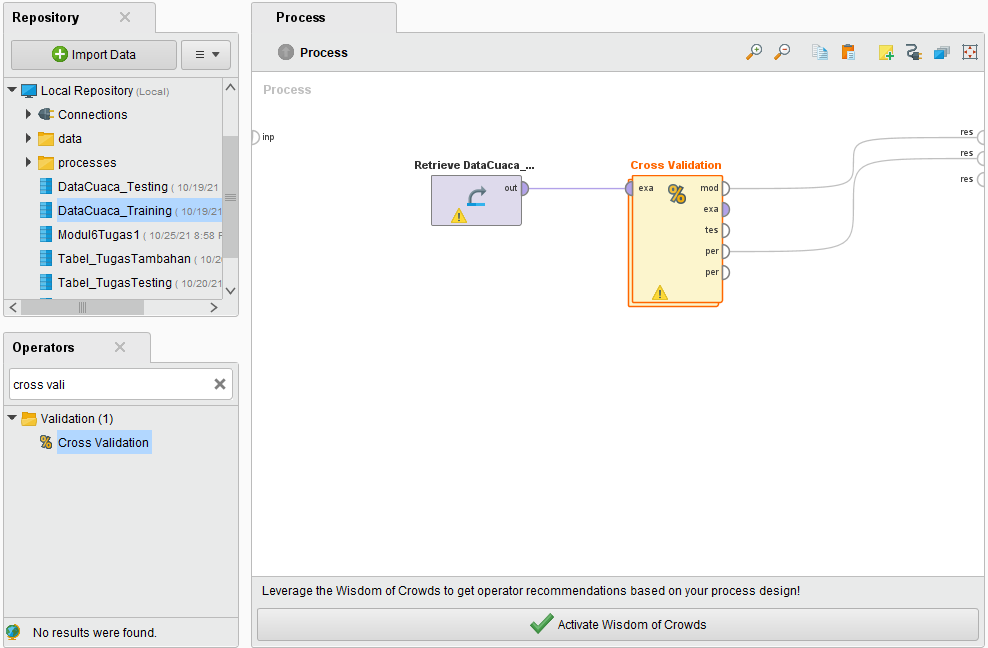
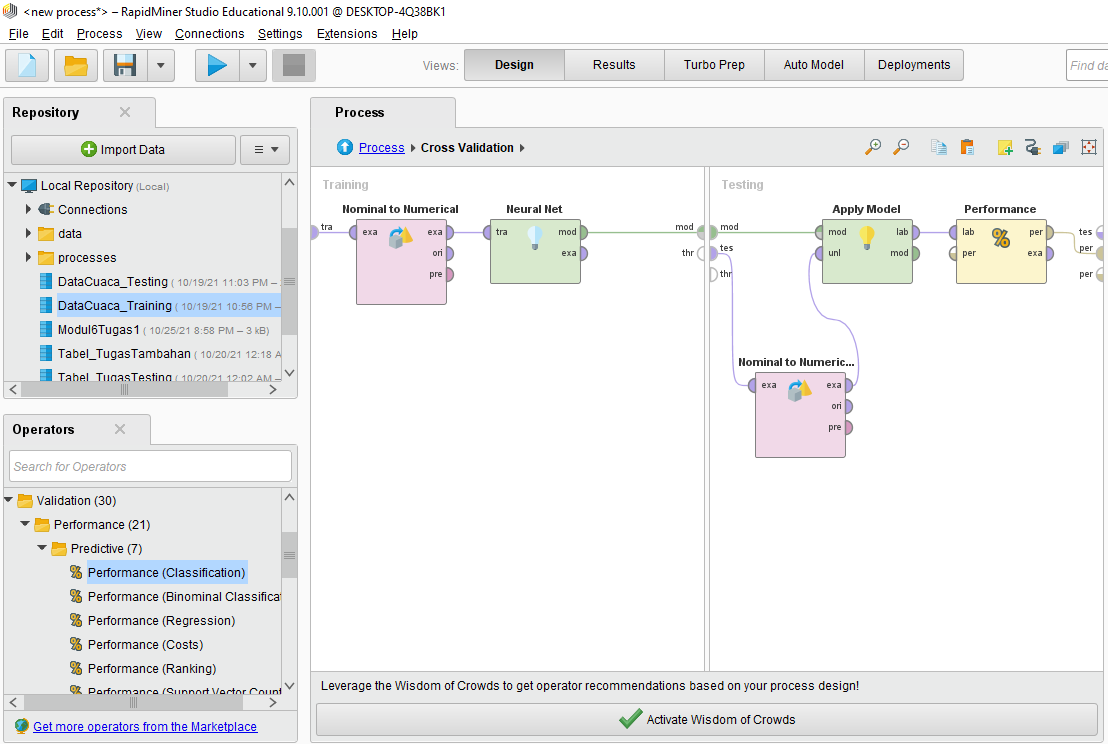
**NAMA : Daffa Putra Alwansyah   
 NIM : L200190031**

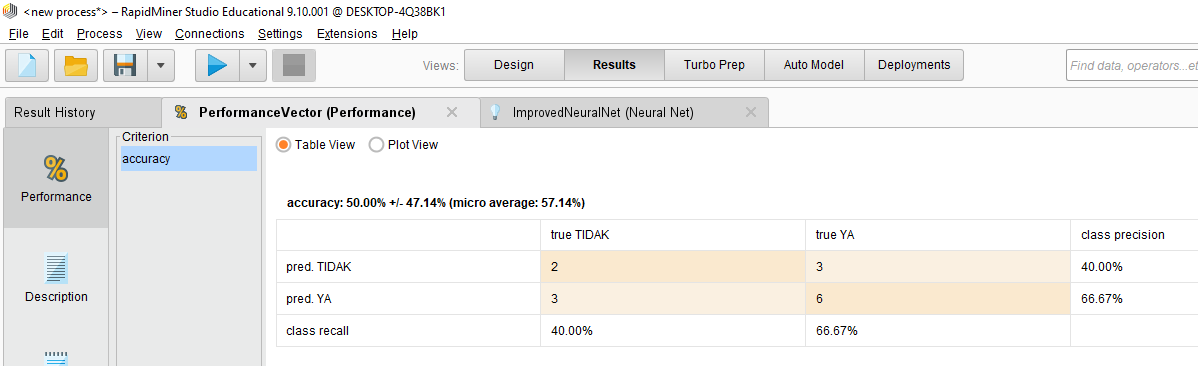
**KELAS : B**

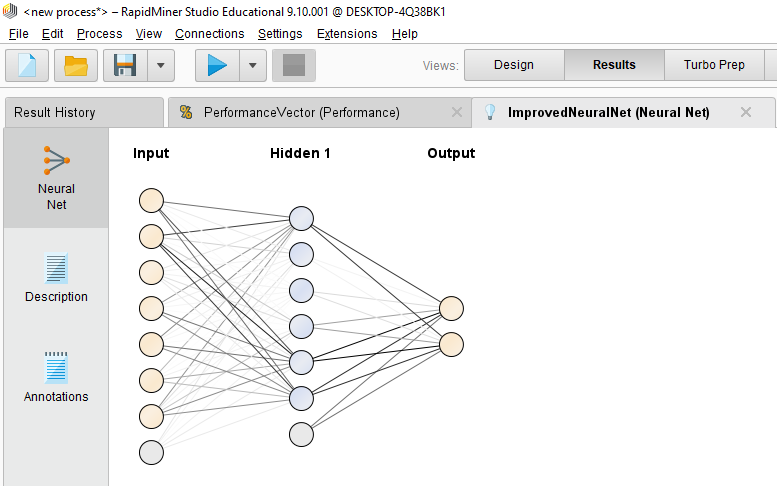
**PRODI : INFORMATIKA**

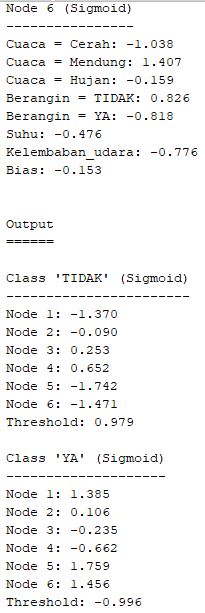
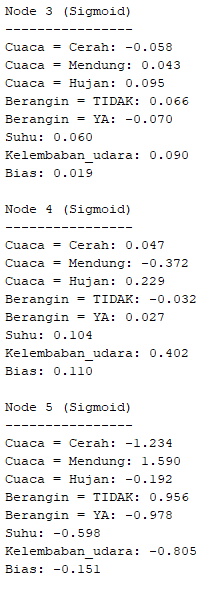
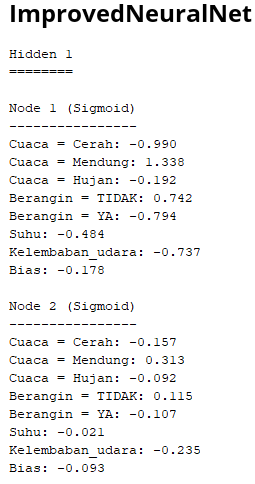
**Fakultas Komunikasi dan Informatika   
Universitas Muhammadiyah Surakarta**

**4.1 Prediksi nilai atribut dengan Neuron Perceptron**1. Buka aplikasi RapidMiner, Drag DataCuaca\_Training dan Data\_CuacaTesting, beserta operatornya yaitu Perceptron dan Apply Model.  
  
2. Setelah itu ada notifikasi Error, yang menyatakan nilai kelas data dalam data Training yang bertipe polynomial tidak dapat diproses, tutup pesan tersebut.  


3. Klik pada Perceptron, lalu “*Help me solve the problem*”  
  
  
4. Setelah itu muncul “Nominal to Numerical” pada Training, tambahkan juga pada Testing.  
  
  
  
  
  
  
  
  
  
  
5. Akan ditunjukan hasil prediksi data testing menggukan JST(Perceptron).  
  
(Hasil prediksi menunjukkan bahwa semua data akan memiliki kelas bermain\_tenis = TIDAK, berdasarkan perbandingan tingkat confidence masing-masing nilai kelas YA dan TIDAK.)  
  
**4.2 Mengetahui Nilai performance Vector pada jaringan saraf tiruan**1. Menggunakan DataCuaca\_Training dan operator Cross Validation, hubungkan port sesuai di modul, setelah itu klik 2x pada Cross Validation.  
  
  
  
  
  
  
2. Cari Operator untuk disebelah kiri yaitu Nominal to Numerical & Neural Net, sedangkan disebelah kanan yaitu Apply Model, Performance dan Nominal to Numerical, setelah itu process dan klik tombol RUN.  
  
  
3. Hasilnya:

* Tab PerformanceVector(Performance), untuk menunjukkan tingkat akurasi, presisi, recall dan lain-lain dalam bentuk tabel dan plot view.  
  
* Tab ImprovedNeuralNet (Neural Net), memperlihatkan arsitektur jaringan syaraf yang digunakan untuk proses. Ada 2 mode view yaitu:

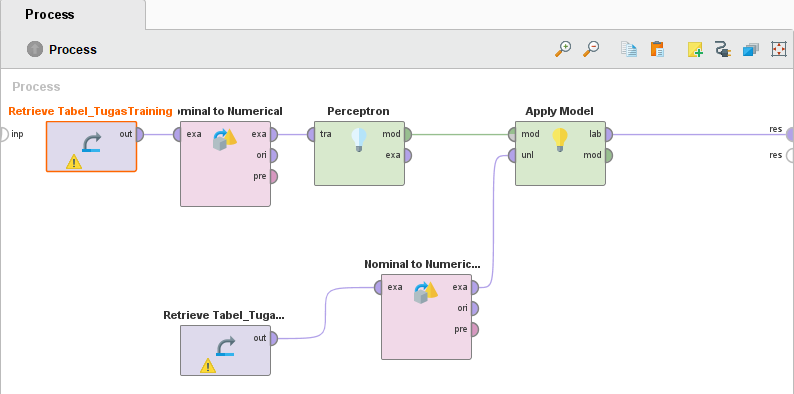
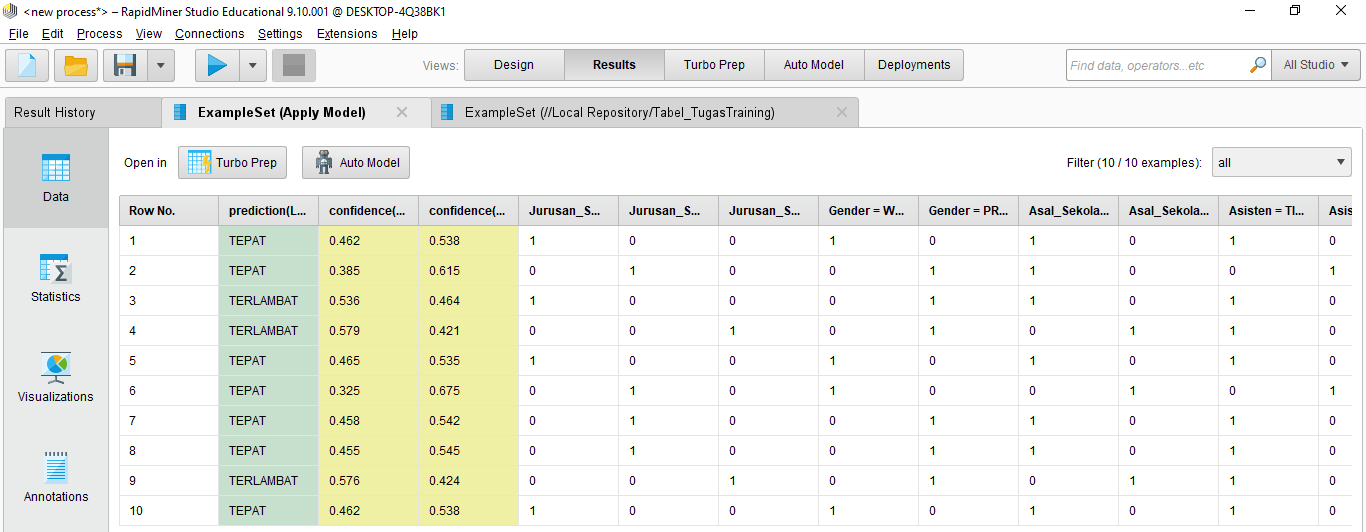
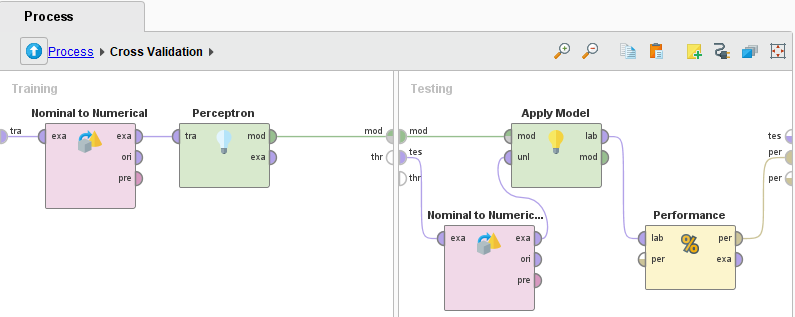
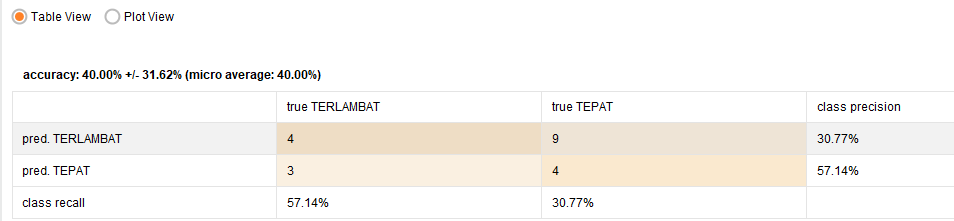
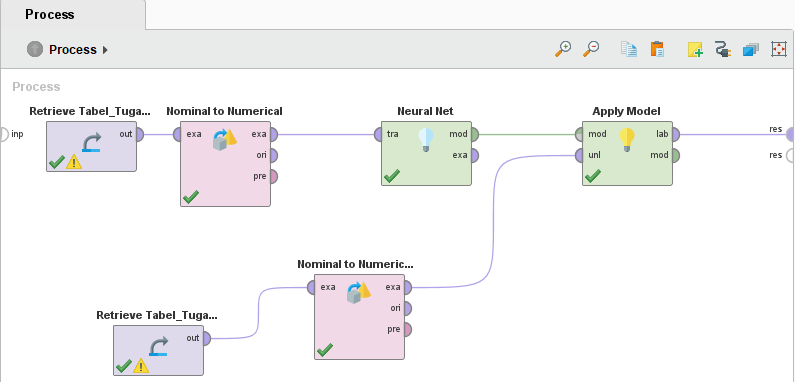
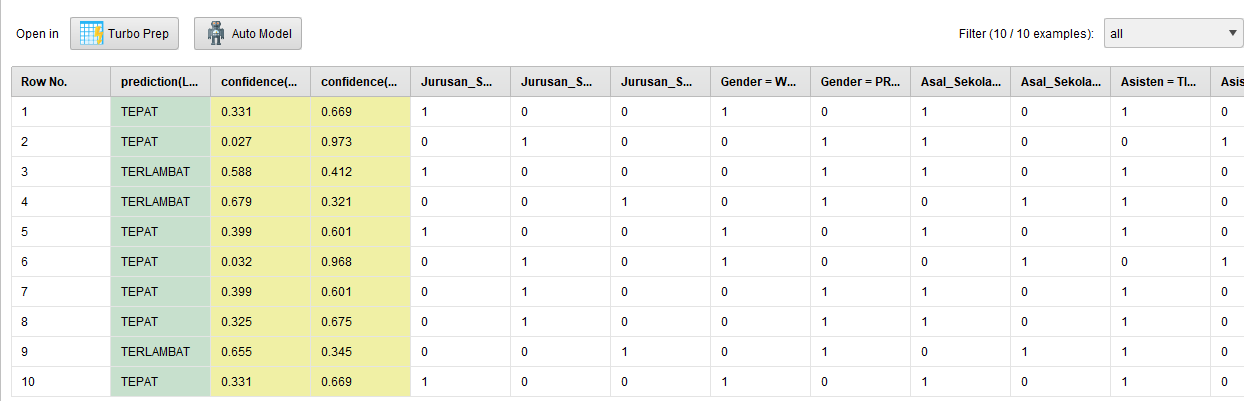
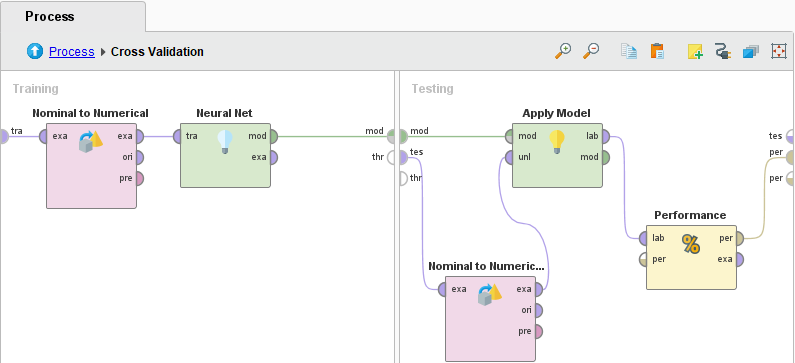
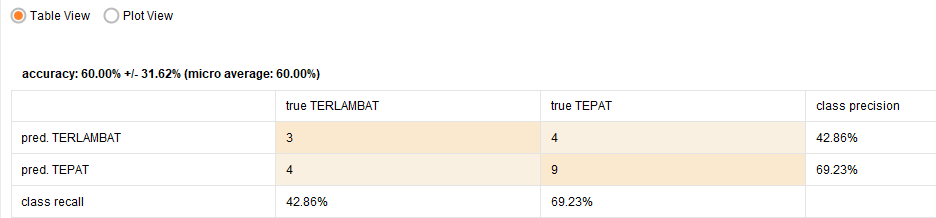
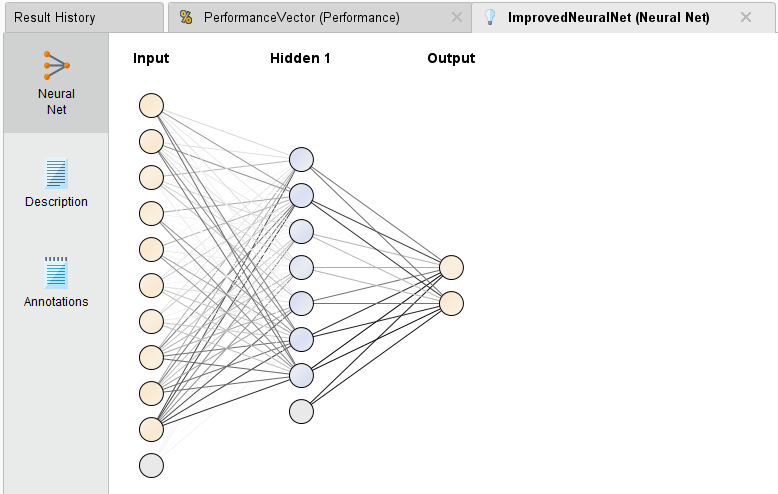
1. Neural Net, digunakan untuk melihat bentuk arsitektur JST  
     
   Berdasarkan arsitektur yang terbentuk, dapat dilihat bahwa jumlah node (simpul) masing-masing layer sebagai berikut :   
   1. Input Layer = 7 node input, dan 1 node berbobot 1   
   2. Hidden Layer = 6 node hidden, dan 1 node berbobot 1   
   3. Output Layer = 2 node (YA, TIDAK)
2. Description, digunakan untuk melihat nilai bobot sigmoid masing-masing node (simpul) pada hidden layer dan output layer.

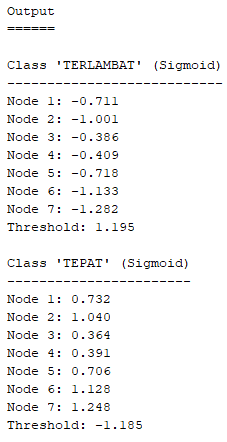
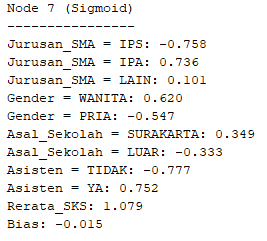
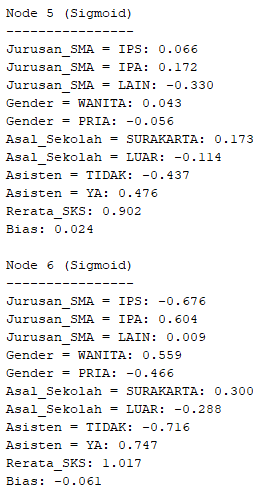
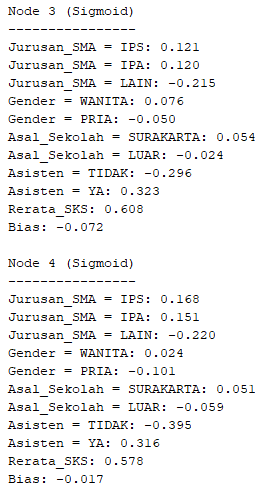
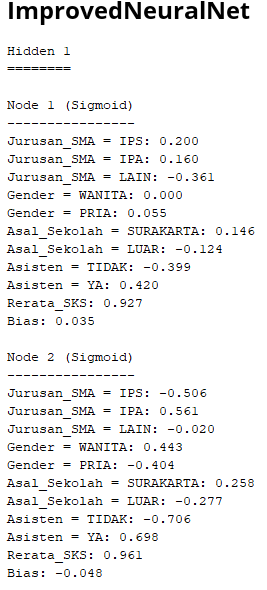






**TUGAS!**

* Menggunakan file exel Tugas nomor 1 Modul 6 (Training) & Tugas nomor 1 Modul 8 (Testing), lalu masukan operator Nominal to Numerical, Pereptron dan Apply Model.
* Hasil prediksi terhadap data testing lama studi mahasiswa dengan menggunakan model Perceptron.  
  
* Menggunakan PerformanceVector untuk mencari tingkat akurasi, presisi dan recall!  
  
* Hasil dengan menggunakan model Perceptron  
  
* Mengganti operator Perceptron menjadi Neural Net.  
  
* Hasil prediksi terhadap data testing lama studi mahasiswa dengan menggunakan model Neural Net.  
  
* Menggunakan PerformanceVector untuk mencari tingkat akurasi, presisi dan recall!  
  
* Hasil dengan menggunakan model Neural Net.  
  
* Arsitektur Jaringan saraf yang terbentuk  
  
* Berdasarkan arsitektur yang terbentuk, dapat dilihat bahwa jumlah node (simpul) masing-masing layer sebagai berikut :   
   1. Input layer = 10 node input, dan 1 node berbobot 1  
   2. Hidden layer = 7 node hidden, dan 1 node berbobot 1  
   3. Output layer = 2 node (TEPAT dan TERLAMBAT)
* Nilai - nilai bobot sigmoid masing-masing node (simpul) pada hidden layer dan output layer :



* Kesimpulan saya yaitu, adanya perubahan pada akurasi, presisi dan recall, dimana pada Perceptron (40%) lebih kecil dari pada Neural Net (60%), jadi lebih akurat pada Neural Net.