

Penerapan PL/SQL Untuk Menentukan Pola Data Latih Pada Aplikasi Berbasis Neural Network

Tri Hastono¹, Prahenusa Wahyu Ciptadi²

^{1,2}Program Studi Informatika, Universitas PGRI Yogyakarta

e-mail: trihastono@upy.ac.id , nusa@upy.ac.id

Intisari

Stabilitas stok dalam kegiatan bisnis merupakan komponen penting. Salah satu cara untuk menjaga kestabilan stok buah durian adalah dengan memperkirakan jumlah kebutuhan konsumen buah durian. Memprediksi jumlah kebutuhan buah durian pelanggan dapat dilakukan dengan melihat pola yang terbentuk dari transaksi penjualan. Namun jika prediksi dilakukan secara manual maka akan memakan waktu yang lama dan rawan terjadi kesalahan. Solusi untuk mengatasi masalah tersebut adalah diperlukannya suatu alat berupa software yang didalamnya sudah terdapat metode penghitungan yang akurat. Salah satu metode yang dapat digunakan untuk menghitung kebutuhan durian pelanggan adalah jaringan syaraf tiruan Adaline. Dan data latih yang digunakan adalah data penjualan buah durian. Data latih dibagi menjadi 2 kelompok, yaitu data input dan data target. Fokus penelitian ini adalah perancangan pl/sql untuk menentukan pola input data secara otomatis untuk aplikasi prediksi jumlah kebutuhan buah durian pelanggan. Jenis pl/sql yang digunakan dalam penelitian ini adalah *procedure* dan *function*.

Kata kunci—pl/sql, data latih, neural network

Abstract

Stock stability in business activities is an important component. One way to maintain the stability of the stock of durian fruit is to estimate the amount of consumer demand for durian fruit. Predicting the amount of durian fruit customers need can be done by looking at the patterns formed from sales transactions.

Penerapan PL/SQL Untuk Menentukan Pola Data Latih Pada Aplikasi Berbasis Neural Network

(Tri Hastono, Prahenua Wahyu Ciptadi)

However, if the prediction is done manually, it will take a long time and be prone to errors. The solution to overcome this problem is the need for a tool in the form of software in which there is already an accurate calculation method. One method that can be used to calculate customer durian needs is the Adaline artificial neural network. And the training data used is durian fruit sales data. Training data is divided into 2 groups, namely input data and target data. The focus of this research is the pl/sql design to determine the data input pattern automatically for the application of predicting the number of customer's durian fruit needs. The types of pl/sql used in this study are procedure and function.

Keywords—pl/sql, learning data, neural network

PENDAHULUAN

Persediaan barang merupakan komponen penting pada kegiatan bisnis jual beli barang. Persediaan barang memiliki pengaruh besar pada lancarnya bisnis penjualan barang. Hal tersebut terlihat ketika persediaan barang stabil, penjualan barang akan berjalan lancar. Dan sebaliknya, jika persediaan barang tidak dalam keadaan stabil, penjualan barang akan terhambat [1]. Tentu sudah bisa dihitung seberapa besar keuntungan yang didapatkan ketika bisnis berjalan lancar.

Untuk mempertahankan kestabilan persediaan barang, pelaku bisnis harus dapat memprediksi seberapa banyak kebutuhan pelanggan atau pembeli [1,2]. Sebenarnya besarnya permintaan dapat diperkirakan dengan cara melihat pola yang ada pada data penjualan yang lalu. Setelah diketahui pola penjualan, maka bisa menentukan seberapa besar barang yang ada digudang.

Untuk pelaku bisnis dengan penjualan skala relatif kecil tentu mudah untuk menghitung besarnya kebutuhan pelanggan atau pembeli. Pelaku bisnis dengan penjualan skala relative kecil dapat menghitung kebutuhan pelanggan atau pembeli secara manual. Jika dilihat dari segi presisi perhitungan tentu perhitungan tersebut masih diragukan tingkat akurasi. Data penjualan yang relative sedikit mempengaruhi tingkat keakurasian dari perhitungan.

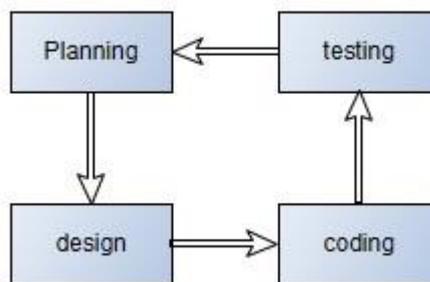
Untuk pelaku bisnis dengan penjualan skala besar tentu akan kesulitan jika perkiraan kebutuhan pelanggan atau pembeli dihitung secara manual [2]. Proses perhitungan manual akan memakan waktu yang lama dan keakurasian perhitungan manual juga masih diragukan. Hal ini dapat semakin parah jika terdapat *human error* dan jenis barangnya terbatas dengan *expired date*. Keuntungan akan berubah menjadi kerugian bisnis [1, 2].

Dari kasus yang diuraikan diatas dapat disimpulkan bahwa para pelaku bisnis membutuhkan tool yang difungsikan untuk memperkirakan kebutuhan pelanggan dengan tingkat akurasi tinggi. Penelitian ini memberikan solusi mengenai perkiraan besarnya kebutuhan pelanggan. Kasus yang dipilih pada penelitian untuk penelitian kasus prediksi jumlah kebutuhan pelanggan buah durian. Penelitian yang diusulkan menggunakan neural network adaline dalam

memprediksi besarnya kebutuhan pelanggan. Fokus dari penelitian adalah perancangan pl/sql untuk otomatisasi pola input pada jaringan neural network adaline. Adapun jenis pl/sql yang digunakan adalah *function* dan *procedure* [].

METODE PENELITIAN

Metode perancangan pl/sql pada penelitian ini menggunakan metode Agile [3]. Jenis metode agile yang dipilih pada penelitian ini adalah metode extreme programming. Extreme Programming dikenalkan oleh Kent Beck pada sebuah project yang ditanganinya. Proses yang ada pada extreme programming mirip dengan waterfall yang meliputi *planning*, *design*, *coding*, and *testing* [3,4].



Gambar 1 Proses metode Extreme programming

1. *Planning*

Pada tahap ini diawali dari mempelajari konteks bisnis software yang diberikan oleh pelanggan dan pencatatan mengenai fungsionalitas dari software. Pada tahap ini juga ditentukan anggota team dan langkah-langkah untuk mewujudkan perangkat lunak yang diinginkan.

2. *Design*

Tahap design ini merupakan tahap lanjutan dari tahap *planning*. Pada tahap ini dilakukan proses penggambaran skema software. Pada tahap ini juga sering

disebut tahap pembuatan blueprint software. Pada tahap ini masih terjadi komunikasi antara pembuat software dan pelanggan.

3. *Coding*

Tahap *coding* adalah salah satu tahap yang dilakukan untuk mengimplementasikan skema yang sudah tergambar dari software yang dipesan.

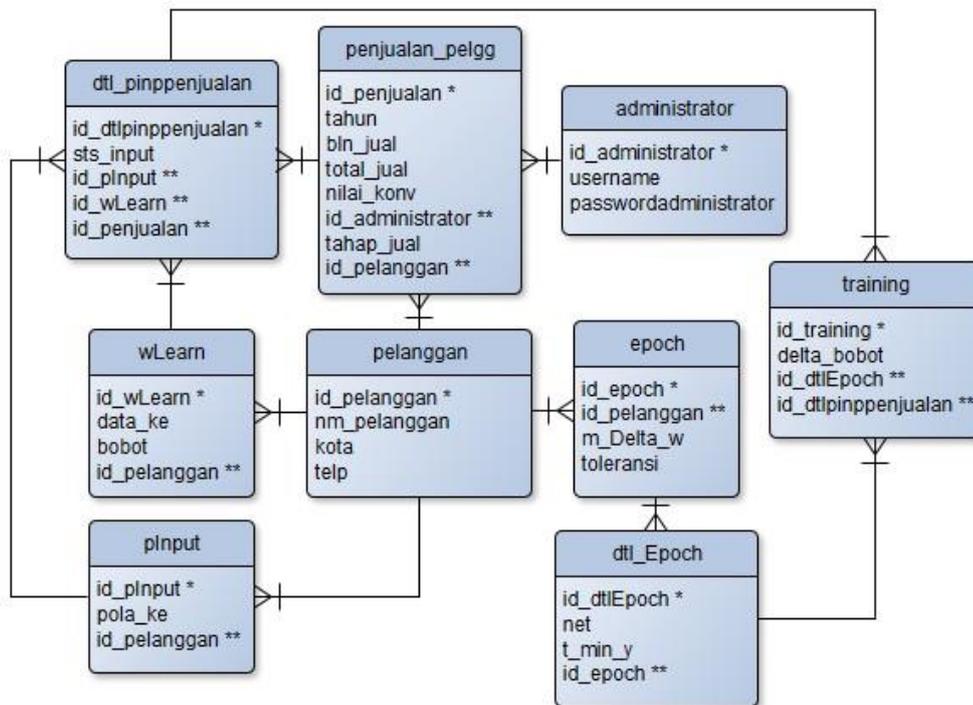
4. *Testing*

Pada tahap *testing* inilah dilakukan uji coba mengenai software yang dipesan oleh pelanggan. Semua fitur dan fungsionalitas dari software diuji pada tahap testing. Hasil dari tahap testing yang menentukan apakah dilakukan evaluasi dari tahap-tahap sebelumnya atau tidak.

HASIL DAN PEMBAHASAN

Pada penelitian ini pola input berasal dari data penjualan yang tersimpan pada basis data. Penentuan pola input secara otomatis diproses ketika operasi tambah data salah satu table dalam basis data. Proses tambah data pada table yang mentrigger aplikasi untuk menentukan pola input adalah *function* yang dirancang khusus menggunakan *store procedure* MySQL.

Basis data pada penelitian ini dirancang menggunakan MySQL Server 5.0 dan platform yang digunakan adalah android. Table pada penelitian ini ada 9, yaitu pelanggan, penjualan_pelgg, wlearn, pinput, dtl_pinpenjualan, administrator, epoch, dtl_Epoch, and training. Untuk relasi table basis data dapat dilihat pada gambar 2 dibawah.



Gambar 2 Relasi table pada basisdata penelitian

Untuk keterangan dari setiap table yang ada pada basis data aplikasi sebagai berikut :

- Table pelanggan

Table pelanggan adalah table yang difungsikan untuk menyimpan data pelanggan buah durian.

- Table penjualan_pelgg

Table penjualan_pelgg adalah table yang difungsikan untuk menyimpan data penjualan dari setiap pelanggan.

- Table wlearn

Table wlearn adalah table yang digunakan untuk menyimpan bobot dari setiap data pola input.

- Table pinput

Table pinput adalah table yang digunakan untuk menyimpan data pola input setiap pelanggan buah durian.

- Table dtl_pinpenjualan

Table dtl_pinpenjualan adalah table yang digunakan untuk menyimpan data detail pola input dari pelanggan buah durian.

- Table Administrator

Table administrator adalah table yang digunakan untuk menyimpan data administrator.

- Table epoch

Table epoch adalah table yang digunakan untuk menyimpan iterasi learning jaringan adaline.

- Table dtl_Epoch

Table dtl_epoch adalah table yang digunakan untuk menyimpan data detail transaksi table epoch

- Table training

Table training adalah table yang digunakan untuk menyimpan data transaksi training jaringan adaline.

Terdapat 2 store procedure yang digunakan pada penelitian ini, yaitu procedure dan function [5,6]. Procedure digunakan untuk membuat data pola input secara otomatis dan function digunakan untuk triger otomatisasi pola input. Proses untuk otomatisasi dari penentuan pola input dilakukan ketika proses tambah data pada table penjualan_pelgg.

Procedure inp_POLAINPUT adalah procedure yang dirancang untuk melakukan pembuatan pola input jaringan Adaline secara otomatis. Algoritma dari procedure inp_POLAINPUT adalah :

- Hapus data table wLearn dan isi table pInput dengan id_pelanggan tertentu.
- Hitung jumlah data penjualan pelanggan yang dipilih dan simpan pada sebuah variable (misalnya cRecord).

Penerapan PL/SQL Untuk Menentukan Pola Data Latih Pada Aplikasi Berbasis Neural Network

(Tri Hastono, Prahonusa Wahyu Ciptadi)

- Hitung jumlah pola input dengan cara membagi cRecord dengan 2 dan simpan pada sebuah variable (misalnya cLoopIndex1).
- Hitung indeks paling atas untuk satu pola input dengan cara hitung cRecord modulus 2. Jika hasilnya adalah 0, maka variable cLoopIndex1 + 1. Dan jika hasilnya adalah ganjil, maka : variable cLoopIndex1 + 2. Hasil proses penjumlahan disimpan pada sebuah variable (misalnya i_top).
- Selama indeks data pertama data penjualan pelanggan < cLoopIndex1, lakukan :
 - Insert data ke table pInput.
 - Selama indeks data jumlah data 1 pola input < i_top, lakukan :
 - Insert into dtl_pinppenjualan dengan data penjualan.
 - Selesai

Adapun kode *query* untuk procedure inp_POLAINPUT dapat dilihat pada gambar 3 dibawah.

```
124 drop procedure if exists inp_POLAINPUT;
125 DELIMITER $$
126 CREATE procedure inp_POLAINPUT(id_plg varchar(100))
127 BEGIN
128     DECLARE ket varchar(4) DEFAULT "I";
129     DECLARE cRecord smallint;
130     DECLARE cLoopIndex1 smallint;
131     DECLARE cLoopIndex2 smallint default 0;
132     DECLARE cLoopIndex3 smallint default 0;
133     DECLARE i_top smallint default 0;
134     DECLARE i_bottom smallint default 0;
135     DECLARE suffixs smallint default 0;
136     DECLARE wIndex smallint default 0;
137
138     delete from wLearn where id_pelanggan=id_plg;
139     delete from pInput where id_pelanggan=id_plg;
140     set cRecord=(select count(*) from penjualan_pelgg where id_pelanggan=id_plg);
141     set cLoopIndex1=cRecord div 2;
142     if cRecord mod 2 = 0 then
143         set i_top=cLoopIndex1+1;
144     else
145         set i_top=cLoopIndex1+2;
146     end if;
147
148     wData: LOOP
149         set wIndex=wIndex+1;
150         insert into wLearn values(concat('w',id_plg,wIndex),wIndex,'0',id_plg);
151         IF wIndex = i_top THEN
152             begin
153                 insert into wLearn values(concat('w',id_plg,wIndex+1),'bias','0',id_plg);
154                 LEAVE wData;
155             end;
156         END IF;
```

```
157 END LOOP wData;
158
159 lData1: LOOP
160     set cLoopIndex2=cLoopIndex2+1;
161     set i_bottom=cLoopIndex2-1;
162     insert into pInput values(concat('PI',id_plg,cLoopIndex2),cLoopIndex2,id_plg);
163     set ket='I';
164     lData2: LOOP
165         set cLoopIndex3=cLoopIndex3+1;
166         set suffixs=suffixs+1;
167         if suffixs mod i_top = 0 then
168             set ket='I';
169         else
170             set ket='I';
171         end if;
172         insert into dtl_pinpenjualan values(concat('DPI',suffixs),ket,concat('PI',id_plg,cLoopIndex2),concat('w',id_plg,cLoopIndex3)
173         (select distinct(id_penjualan)from penjualan_pelgg order by (right(id_penjualan,length(id_penjualan)-2)*1) limit i_bottom,1))
174         set i_bottom=i_bottom+1;
175         IF cLoopIndex3 = i_top THEN
176             begin
177                 set cLoopIndex3=0;
178             end;
179             LEAVE lData2;
180         END IF;
181     ITERATE lData2;
182 END LOOP lData2;
183 IF cLoopIndex2 = cLoopIndex1 THEN
184     LEAVE lData1;
185 END IF;
186 END LOOP lData1;
187 -END**
188 DELIMITER ;
```

Gambar 3. Kode program procedure inp_POLAINPUT

KESIMPULAN

Dari hasil perancangan store procedure untuk input pola data latih pada aplikasi memberikan beberapa kemudahan pada proses perancangan maupun pengembangan. Hal tersebut dikarenakan store procedure yang dibuat ditanam pada sisi server, sehingga sisi client hanya unggah data dan tampil data.

DAFTAR PUSTAKA

- [1] Ferawati, Davita Fersiartha, K., & Yuliana, I. (2020). Analisis Pengaruh Persediaan Barang Dan Penjualan Terhadap Laba Perusahaan (Studi Kasus Cv Davin Jaya Karimun). *Jurnal Cafeteria*, 1(2), 33–44.
- [2] Priatna, H., & Trisnawan, M. R. (2016). PENGARUH PERSEDIAAN BAHAN BAKU DAN VOLUME PENJUALAN TERHADAP LABA BERSIH PERUSAHAAN (STUDI KASUS PADA CV. CISATEX DI DAERAH MAJALAYA). *AKURAT*, 7(3), 55–64.
- [3] Fojtik, R. (2011). Extreme programming in development of specific software. *Procedia Computer Science*, 3, 1464–1468. <https://doi.org/10.1016/j.procs.2011.01.032>
- [4] Suryantara, I. G. N., & Andry, J. F. (2018). Development of Medical Record With Extreme Programming SDLC. *International Journal of New Media Technology*, 5(1), 47–53. <https://doi.org/10.31937/ijnmt.v5i1.706>

Penerapan PL/SQL Untuk Menentukan Pola Data Latih Pada Aplikasi Berbasis Neural Network

(Tri Hastono, Prahenua Wahyu Ciptadi)

- [5] Septiadi, A. D., & Bae, L. J. (2020). Comparative Analysis of Database Query Storage Performance Between Stored Procedure and Function. *IJIS: International Journal of Informatics and Information Systems*, 3(2), 60–66. <https://doi.org/10.47738/ijjis.v3i2.66>
- [6] Fathulloh, A. H., & Adauwiyah, H. I. (2021). Perbandingan Tingkat Efisiensi Waktu Query Select Pada Database Interface Navicat Dan Sqlyog Di Mysql Dbms. *Applied Information Systems and Management (AISM)*, 4(2), 101–105.