

Implementasi regresi linear untuk prediksi penjualan pada aplikasi point of sales restoran

by John Doe

Submission date: 24-Nov-2020 01:54PM (UTC+0700)

Submission ID: 1454601195

File name: jurnal_automata_AutoRecovered_1.2_5.docx (171.21K)

Word count: 996

Character count: 6700

Implementasi regresi linear untuk prediksi penjualan pada aplikasi point of sales restoran

Abstrak—Dalam data transaksi restoran sebenarnya terdapat berbagai macam informasi penting yang masih tersebar dan tersembunyi yang seharusnya dapat digunakan untuk berbagai kepentingan strategi bisnis restoran seperti penjualan. Manajer restoran harus mengetahui makanan dan minuman yang lebih laris dengan memprediksi penjualan berdasarkan data penjualan bulan-bulan sebelumnya karena itu diperlukan sebuah algoritma untuk melakukan prediksi.

Regresi linier merupakan sebuah algoritma yang memodelkan hubungan antara suatu variabel terikat dan variabel bebas yang dapat digunakan untuk melakukan peramalan berdasarkan data-data sebelumnya. Tujuan utamanya adalah untuk membantu manajer restoran untuk membuat keputusan yang tepat berdasarkan hasil prediksi dan data yang ada.

kata kunci : Regresi linier, Prediksi, Penjualan

I. Pendahuluan

Pada Suatu restoran proses transaksi merupakan nyawa dari restoran sehingga penting untuk dilakukan penggalan informasi dari data transaksi tersebut karena informasi dapat berguna sebagai tumpuan manajer restoran mengambil keputusan bisnis seperti melihat peluang dan pengalokasian sumber daya. Selama ini kebanyakan proses transaksi restoran masih manual dalam pencatatannya sehingga memakan waktu dan memungkinkan terjadinya kesalahan dimana bagian kasir dan dapur salah mengira pesanan sehingga makanan yang dibuat tidak sesuai pesanan pelanggan secara tidak langsung membuang sumberdaya dengan percuma dengan menggunakan point of sales dapat mengecek terlebih dahulu untuk ketersediaan menu tersebut dan mengurangi kemungkinan *human error* dalam transaksi [1].

Data transaksi yang ada di dalam point of sales masih harus dikelola lagi untuk mendapatkan informasi yang

bisa membantu pengambilan strategi bisnis. Dari data transaksi dapat diprediksi penjualan menu apa yang laris dan kurang laris sehingga dapat meminimalkan sumber daya untuk menu menu yang kurang laris dan memaksimalkan penjualan menu yang laris sehingga dapat membantu efisiensi restoran. Maka dibutuhkan sebuah algoritma yang dapat memprediksi penjualan restoran pada masa depan. Prediksi dapat dimanfaatkan untuk memperkirakan penjualan restoran. Prediksi dilakukan dengan membuat peramalan untuk melihat nilai di masa depan dengan nilai di masa lalu [2].

Regresi linear adalah hubungan model secara linier antara variabel dependen dan variabel bebas untuk memprediksi nilai dari dependen jika variabel bebas mengalami perubahan nilai [3]. Algoritma ini yang akan memprediksi penjualan pada restoran sehingga kita bisa memprediksi Penjualan bulan berikutnya agar restoran dapat menentukan strategi untuk lebih mengefektifkan pengalokasian sumber daya yang dimiliki restoran.

II. Kajian pustaka

A. Regresi Linier Untuk Prediksi Penjualan Properti

Pada Penelitian yang dilakukan Ghebyla Najla Ayuni dan Devi Fitriana Yang berjudul "Penerapan metode regresi linear untuk prediksi property pada PT XYZ", mereka melakukan prediksi penjualan properti dengan mengembangkan hubungan antara variabel dependen atau variabel akibat (Y) yang dipengaruhi oleh variabel independent atau sebab (X) menyimpulkan peramalan penjualan yang menggunakan algoritma prediksi regresi linear dapat dikatakan cukup baik berdasarkan hasil pengujian MSE, RMSE, dan MAPE [4].

B. Regresi Linear untuk Prediksi Penjualan Batik

Penelitian yang dilakukan oleh Tutik Khotimah dan Ratih Nindiyasari yang berjudul "Forecasting Dengan Metode Regresi Linier Pada Sistem Penunjang Keputusan Untuk Memprediksi Jumlah Penjualan

Batik (Studi Kasus Kub Sarwo Endah Batik tulis Lasem)", Metode regresi linier terbukti dapat memprediksi penjualan dengan baik berdasarkan data penjualan di masa lalu[2].

C. Point Of Sales

Point of sales adalah sebuah aplikasi yang ditujukan untuk mencatat proses transaksi dari pemesanan dan pembayaran[5].

D. Codeigniter

Merupakan framework yang dikembangkan oleh komunitas open source yang memisahkan antara data dan tampilan sehingga memungkinkan pengembangan web dengan cepat serta memudahkan pengolahan web tersebut[6].

III. Metodologi

- Pengumpulan Data**
Pengumpulan Data Transaksi pada restoran XYZ selama periode Januari 2020-Desember 2020 atau selama kurun waktu setahun.
- Analisa Kebutuhan Sistem**
Analisa kebutuhan dilakukan untuk mendapatkan spesifikasi sistem yang akan dibangun agar sesuai dengan kebutuhan transaksi dan prediksi restoran.
- Perancangan sistem**
Tahap perancangan pada tahap ini dilakukan untuk merancang sistem prediksi pada point of sales menggunakan algoritma Regresi linear.
- Implementasi**
Tahap implementasi pada tahap ini dilakukan untuk membangun sistem yang telah dirancang. dilakukan implementasi setelah Analisa kebutuhan dan perancangan sistem selesai dilakukan. Pada tahap ini akan dibuat prediksi penjualan menu dan penjualan total perbulan nya lalu implementasi prediksi nya pada aplikasi point of sales.
- Pengujian**
Tahap ini untuk menguji hasil pada prediksi menggunakan regresi linier

IV. Hasil dan Pembahasan

A. perhitungan Regresi linier

Dari data transaksi yang dilakukan restoran transaksi penjualan bulanan, penjualan bulan per kategori seperti tabel dibawah

Tabel 1 :Penjualan bulanan

No	bulan	Penjualan
1	Januari	400
2	febuari	450
3	maret	370
4	april	420
5	mei	420
6	juni	460

Tabel 1.2 :Perhitungan x, y, xx, dan xy

No	bulan	Penjualan	x	y	xx	xy
1	Januari	400	1	400	1	400
2	febuari	450	2	450	4	900
3	maret	370	3	370	9	1110
4	april	420	4	420	16	1680
5	mei	420	5	420	25	2100
6	juni	460	6	460	36	2760
Total			21	2520	91	8950

Rumus Regresi linier

$$y = b_0 + b_1x$$

Rumus mencari b0 dan b1

$$b_1 = \frac{n\sum xy - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

Maka b0 dan b1 dari tabel diatas adalah

$$b_0 = 394$$

$$b_1 = 7.43$$

Sehingga didapatkan rumus regresi linier untuk tabel 1 adalah

$$y = 394 + 7.43x$$

Perhitungan untuk regresi data per kategori

Implementasi code

Tabel 2 penjualan makanan

No	bulan	makanan x	y	xx	xy
1	januari	130	1	130	1
2	febuari	150	2	150	4
3	maret	110	3	110	9
4	april	135	4	135	16
5	mei	134	5	134	25
6	juni	200	6	200	36
	Total		21	859	91
					3170

Hasil pencarian b0 dan b1

$$b_0 = 110.47$$

$$b_1 = 9.34$$

Regresi liniernya

$$y = 110.47 + 9.34x$$

Tabel 3 penjualan minuman

bulan	minuman x	y	xx	xy
januari	270	1	270	1
febuari	300	2	300	4
maret	260	3	260	9
april	285	4	285	16
mei	286	5	286	25
juni	260	6	260	36
total		21	1661	91
				5780

Hasil pencarian b0 dan b1

$$b_0 = 283.53$$

$$b_1 = -1.91$$

Regresi linier nya

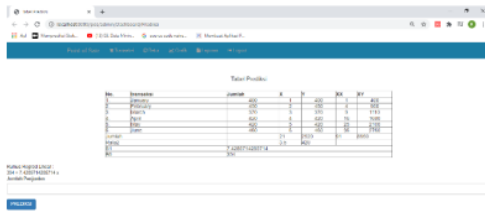
$$y = 283.53 + -1.91x$$

```

1 # Importing the necessary libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from sklearn.linear_model import LinearRegression
6
7 # Creating a dataset for food sales
8 data_food = {
9     'Month': ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni'],
10    'Food': [130, 150, 110, 135, 134, 200],
11    'Sales': [1, 2, 3, 4, 5, 6]
12}
13 df_food = pd.DataFrame(data_food)
14
15 # Creating a dataset for beverage sales
16 data_bev = {
17     'Month': ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni'],
18     'Beverage': [270, 300, 260, 285, 286, 260],
19     'Sales': [1, 2, 3, 4, 5, 6]
20}
21 df_bev = pd.DataFrame(data_bev)
22
23 # Fitting a linear regression model for food sales
24 X_food = df_food['Food'].values.reshape(-1, 1)
25 y_food = df_food['Sales'].values
26 model_food = LinearRegression()
27 model_food.fit(X_food, y_food)
28
29 # Predicting food sales for a given month
30 month = 'Januari'
31 X_pred = df_food[df_food['Month'] == month]['Food'].values.reshape(-1, 1)
32 y_pred_food = model_food.predict(X_pred)
33
34 # Fitting a linear regression model for beverage sales
35 X_bev = df_bev['Beverage'].values.reshape(-1, 1)
36 y_bev = df_bev['Sales'].values
37 model_bev = LinearRegression()
38 model_bev.fit(X_bev, y_bev)
39
40 # Predicting beverage sales for a given month
41 month = 'Januari'
42 X_pred = df_bev[df_bev['Month'] == month]['Beverage'].values.reshape(-1, 1)
43 y_pred_bev = model_bev.predict(X_pred)
44
45 # Printing the results
46 print("Predicted food sales for January: ", y_pred_food)
47 print("Predicted beverage sales for January: ", y_pred_bev)
48
49 # Plotting the data
50 plt.figure(figsize=(10, 5))
51 plt.title("Food Sales Data")
52 plt.xlabel("Food")
53 plt.ylabel("Sales")
54 plt.scatter(X_food, y_food)
55 plt.grid(True)
56 plt.show()
57
58 plt.figure(figsize=(10, 5))
59 plt.title("Beverage Sales Data")
60 plt.xlabel("Beverage")
61 plt.ylabel("Sales")
62 plt.scatter(X_bev, y_bev)
63 plt.grid(True)
64 plt.show()
65
66 # Calculating the regression coefficients
67 print("Coefficients for food sales: ", model_food.coef_)
68 print("Coefficients for beverage sales: ", model_bev.coef_)
69
70 # Calculating the intercepts
71 print("Intercept for food sales: ", model_food.intercept_)
72 print("Intercept for beverage sales: ", model_bev.intercept_)
73
74 # Calculating the R-squared values
75 print("R-squared for food sales: ", model_food.score(X_food, y_food))
76 print("R-squared for beverage sales: ", model_bev.score(X_bev, y_bev))
77
78 # Calculating the Mean Squared Error (MSE)
79 print("MSE for food sales: ", np.mean((y_food - model_food.predict(X_food))**2))
80 print("MSE for beverage sales: ", np.mean((y_bev - model_bev.predict(X_bev))**2))
81
82 # Calculating the Mean Absolute Error (MAE)
83 print("MAE for food sales: ", np.mean(np.abs(y_food - model_food.predict(X_food))))
84 print("MAE for beverage sales: ", np.mean(np.abs(y_bev - model_bev.predict(X_bev))))
85
86 # Calculating the Root Mean Squared Error (RMSE)
87 print("RMSE for food sales: ", np.sqrt(np.mean((y_food - model_food.predict(X_food))**2)))
88 print("RMSE for beverage sales: ", np.sqrt(np.mean((y_bev - model_bev.predict(X_bev))**2)))
89
90 # Calculating the Adjusted R-squared values
91 print("Adjusted R-squared for food sales: ", model_food.score(X_food, y_food) * (len(X_food) - 1) / (len(X_food) - 2))
92 print("Adjusted R-squared for beverage sales: ", model_bev.score(X_bev, y_bev) * (len(X_bev) - 1) / (len(X_bev) - 2))
93
94 # Calculating the F-statistic
95 print("F-statistic for food sales: ", model_food.fstat_)
96 print("F-statistic for beverage sales: ", model_bev.fstat_)
97
98 # Calculating the p-value
99 print("p-value for food sales: ", model_food.pvalues_)
100 print("p-value for beverage sales: ", model_bev.pvalues_)
101
102 # Calculating the confidence interval
103 print("Confidence interval for food sales: ", model_food.conf_int())
104 print("Confidence interval for beverage sales: ", model_bev.conf_int())
105
106 # Calculating the prediction interval
107 print("Prediction interval for food sales: ", model_food.predict_interval())
108 print("Prediction interval for beverage sales: ", model_bev.predict_interval())
109
110 # Calculating the variance-covariance matrix
111 print("Variance-covariance matrix for food sales: ", model_food.cov_params_)
112 print("Variance-covariance matrix for beverage sales: ", model_bev.cov_params_)
113
114 # Calculating the standard error of the coefficients
115 print("Standard error for food sales: ", model_food.stderr_)
116 print("Standard error for beverage sales: ", model_bev.stderr_)
117
118 # Calculating the t-statistic
119 print("t-statistic for food sales: ", model_food.tvalues_)
120 print("t-statistic for beverage sales: ", model_bev.tvalues_)
121
122 # Calculating the probability density function (PDF)
123 print("PDF for food sales: ", model_food.pdf(X_food))
124 print("PDF for beverage sales: ", model_bev.pdf(X_bev))
125
126 # Calculating the cumulative distribution function (CDF)
127 print("CDF for food sales: ", model_food.cdf(X_food))
128 print("CDF for beverage sales: ", model_bev.cdf(X_bev))
129
130 # Calculating the log-likelihood function
131 print("Log-likelihood for food sales: ", model_food.log_likelihood_)
132 print("Log-likelihood for beverage sales: ", model_bev.log_likelihood_)
133
134 # Calculating the Akaike Information Criterion (AIC)
135 print("AIC for food sales: ", model_food.aic_)
136 print("AIC for beverage sales: ", model_bev.aic_)
137
138 # Calculating the Bayesian Information Criterion (BIC)
139 print("BIC for food sales: ", model_food.bic_)
140 print("BIC for beverage sales: ", model_bev.bic_)
141
142 # Calculating the Schwarz Criterion (SC)
143 print("SC for food sales: ", model_food.sc_)
144 print("SC for beverage sales: ", model_bev.sc_)
145
146 # Calculating the Hannan-Quinn Criterion (HQ)
147 print("HQ for food sales: ", model_food.hq_)
148 print("HQ for beverage sales: ", model_bev.hq_)
149
150 # Calculating the Akaike weights
151 print("Akaike weights for food sales: ", model_food.aic_weights_)
152 print("Akaike weights for beverage sales: ", model_bev.aic_weights_)
153
154 # Calculating the Bayes factor
155 print("Bayes factor for food sales: ", model_food.bayes_factor_)
156 print("Bayes factor for beverage sales: ", model_bev.bayes_factor_)
157
158 # Calculating the posterior probability
159 print("Posterior probability for food sales: ", model_food.posterior_)
160 print("Posterior probability for beverage sales: ", model_bev.posterior_)
161
162 # Calculating the prior probability
163 print("Prior probability for food sales: ", model_food.prior_)
164 print("Prior probability for beverage sales: ", model_bev.prior_)
165
166 # Calculating the likelihood function
167 print("Likelihood for food sales: ", model_food.likelihood_)
168 print("Likelihood for beverage sales: ", model_bev.likelihood_)
169
170 # Calculating the log-likelihood gradient
171 print("Log-likelihood gradient for food sales: ", model_food.gradient_)
172 print("Log-likelihood gradient for beverage sales: ", model_bev.gradient_)
173
174 # Calculating the Hessian matrix
175 print("Hessian matrix for food sales: ", model_food.hessian_)
176 print("Hessian matrix for beverage sales: ", model_bev.hessian_)
177
178 # Calculating the Fisher information matrix
179 print("Fisher information matrix for food sales: ", model_food.fisher_info_)
180 print("Fisher information matrix for beverage sales: ", model_bev.fisher_info_)
181
182 # Calculating the Cramere-Rao lower bound
183 print("Cramere-Rao lower bound for food sales: ", model_food.cramere_rao_)
184 print("Cramere-Rao lower bound for beverage sales: ", model_bev.cramere_rao_)
185
186 # Calculating the maximum likelihood estimates
187 print("Maximum likelihood estimates for food sales: ", model_food.mle_)
188 print("Maximum likelihood estimates for beverage sales: ", model_bev.mle_)
189
190 # Calculating the method of moments estimates
191 print("Method of moments estimates for food sales: ", model_food.mom_)
192 print("Method of moments estimates for beverage sales: ", model_bev.mom_)
193
194 # Calculating the generalized likelihood ratio test
195 print("GLRT for food sales: ", model_food.glrt_)
196 print("GLRT for beverage sales: ", model_bev.glrt_)
197
198 # Calculating the Wald test
199 print("Wald test for food sales: ", model_food.wald_)
200 print("Wald test for beverage sales: ", model_bev.wald_)
201
202 # Calculating the Lagrange multiplier test
203 print("Lagrange multiplier test for food sales: ", model_food.lagrange_)
204 print("Lagrange multiplier test for beverage sales: ", model_bev.lagrange_)
205
206 # Calculating the likelihood ratio test
207 print("Likelihood ratio test for food sales: ", model_food.lr_)
208 print("Likelihood ratio test for beverage sales: ", model_bev.lr_)
209
210 # Calculating the score test
211 print("Score test for food sales: ", model_food.score_)
212 print("Score test for beverage sales: ", model_bev.score_)
213
214 # Calculating the information criterion
215 print("Information criterion for food sales: ", model_food.info_)
216 print("Information criterion for beverage sales: ", model_bev.info_)
217
218 # Calculating the Akaike weights
219 print("Akaike weights for food sales: ", model_food.aic_weights_)
220 print("Akaike weights for beverage sales: ", model_bev.aic_weights_)
221
222 # Calculating the Bayes factor
223 print("Bayes factor for food sales: ", model_food.bayes_factor_)
224 print("Bayes factor for beverage sales: ", model_bev.bayes_factor_)
225
226 # Calculating the posterior probability
227 print("Posterior probability for food sales: ", model_food.posterior_)
228 print("Posterior probability for beverage sales: ", model_bev.posterior_)
229
230 # Calculating the prior probability
231 print("Prior probability for food sales: ", model_food.prior_)
232 print("Prior probability for beverage sales: ", model_bev.prior_)
233
234 # Calculating the likelihood function
235 print("Likelihood for food sales: ", model_food.likelihood_)
236 print("Likelihood for beverage sales: ", model_bev.likelihood_)
237
238 # Calculating the log-likelihood gradient
239 print("Log-likelihood gradient for food sales: ", model_food.gradient_)
240 print("Log-likelihood gradient for beverage sales: ", model_bev.gradient_)
241
242 # Calculating the Hessian matrix
243 print("Hessian matrix for food sales: ", model_food.hessian_)
244 print("Hessian matrix for beverage sales: ", model_bev.hessian_)
245
246 # Calculating the Fisher information matrix
247 print("Fisher information matrix for food sales: ", model_food.fisher_info_)
248 print("Fisher information matrix for beverage sales: ", model_bev.fisher_info_)
249
250 # Calculating the Cramere-Rao lower bound
251 print("Cramere-Rao lower bound for food sales: ", model_food.cramere_rao_)
252 print("Cramere-Rao lower bound for beverage sales: ", model_bev.cramere_rao_)
253
254 # Calculating the maximum likelihood estimates
255 print("Maximum likelihood estimates for food sales: ", model_food.mle_)
256 print("Maximum likelihood estimates for beverage sales: ", model_bev.mle_)
257
258 # Calculating the method of moments estimates
259 print("Method of moments estimates for food sales: ", model_food.mom_)
260 print("Method of moments estimates for beverage sales: ", model_bev.mom_)
261
262 # Calculating the generalized likelihood ratio test
263 print("GLRT for food sales: ", model_food.glrt_)
264 print("GLRT for beverage sales: ", model_bev.glrt_)
265
266 # Calculating the Wald test
267 print("Wald test for food sales: ", model_food.wald_)
268 print("Wald test for beverage sales: ", model_bev.wald_)
269
270 # Calculating the Lagrange multiplier test
271 print("Lagrange multiplier test for food sales: ", model_food.lagrange_)
272 print("Lagrange multiplier test for beverage sales: ", model_bev.lagrange_)
273
274 # Calculating the likelihood ratio test
275 print("Likelihood ratio test for food sales: ", model_food.lr_)
276 print("Likelihood ratio test for beverage sales: ", model_bev.lr_)
277
278 # Calculating the score test
279 print("Score test for food sales: ", model_food.score_)
280 print("Score test for beverage sales: ", model_bev.score_)
281
282 # Calculating the information criterion
283 print("Information criterion for food sales: ", model_food.info_)
284 print("Information criterion for beverage sales: ", model_bev.info_)
285
286 # Calculating the Akaike weights
287 print("Akaike weights for food sales: ", model_food.aic_weights_)
288 print("Akaike weights for beverage sales: ", model_bev.aic_weights_)
289
290 # Calculating the Bayes factor
291 print("Bayes factor for food sales: ", model_food.bayes_factor_)
292 print("Bayes factor for beverage sales: ", model_bev.bayes_factor_)
293
294 # Calculating the posterior probability
295 print("Posterior probability for food sales: ", model_food.posterior_)
296 print("Posterior probability for beverage sales: ", model_bev.posterior_)
297
298 # Calculating the prior probability
299 print("Prior probability for food sales: ", model_food.prior_)
300 print("Prior probability for beverage sales: ", model_bev.prior_)
301
299

```

Tampilan hasil implementasi



No	Nama	Jumlah	K	Z	M1	M2	M3	M4	M5	M6
1
2
3
4
5
6
7
8
9
10

V. Kesimpulan

Berdasarkan hasil pembahasan diatas dapat diambil kesimpulan :

1. hasil prediksi penjualan restoran menggunakan regresi linier menggunakan pemrograman dapat mendapatkan hasil prediksi dengan lebih cepat dan akurat.

2. dengan mengetahui prediksi penjualan restoran dapat mengambil strategi untuk memaksimalkan sumberdaya yang dimiliki.

Daftar Pustaka

- [1] H. T. H. Herman, S. Rostianingsih, and A. Setiawan, "Pembuatan Aplikasi Point of Sales untuk Rumah Makan Dapur Rinjani," *Infra*, vol. 4, p. 6, 2016, [Online]. Available: <https://media.neliti.com/media/publications/108536-ID-pembuatan-aplikasi-point-of-sales-untuk.pdf>.
- [2] T. Khotimah and R. Nindiyasari, "Forecasting Dengan Metode Regresi Linier Pada Sistem Penunjang Keputusan Untuk Memprediksi Jumlah Penjualan Batik (Studi Kasus Kub Sarwo Endah Batik Tulis Lasem)," *J. Mantik Penusa*, vol. 1, no. 1, pp. 71–92, 2017, [Online]. Available: <http://ejournal.pelitanusantara.ac.id/index.php/mantik/article/view/541>.
- [3] C. Siregar, A. S. Sembiring, and H. K. Siburian, "Perancangan Aplikasi Prediksi Penjualan Laptop Dengan Menerapkan Metode Regresi Linier," *J. Pelita Inform.*, vol. 17, no. 4, pp. 416–421, 2018, [Online]. Available: <https://ejournal.stmik-budidarma.ac.id/index.php/pelita/article/view/1079/932>.
- [4] G. N. Ayuni and D. Fitriah, "Penerapan Metode Regresi Linear Untuk Prediksi Penjualan Properti pada PT XYZ," *J. Telemat.*, vol. 14, no. 2, pp. 79–86, 2019, [Online]. Available: <https://journal.ithb.ac.id/telematika/article/view/321>.
- [5] S. C. Cahyodi and R. W. Arifin, "Sistem Informasi Point Of Sales Berbasis Web Pada Colony Amaranta Bekasi," *Mei Rev. Mei*, vol. 1, no. 12, pp. 189–204, 2017, [Online]. Available: [http://download.portalgaruda.org/article.php?article=535077&val=10928&title= Sistem Informasi Point Of Sales Berbasis Web Pada Colony Amaranta Bekasi](http://download.portalgaruda.org/article.php?article=535077&val=10928&title=Sistem%20Informasi%20Point%20Of%20Sales%20Berbasis%20Web%20Pada%20Colony%20Amaranta%20Bekasi).
- [6] L. Afuan, "Pemanfaatan Framework Codeigniter dalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed," *Juita*, vol. I, no. 2, pp. 39–44, 2010.

Implementasi regresi linear untuk prediksi penjualan pada aplikasi point of sales restoran

ORIGINALITY REPORT

15%

SIMILARITY INDEX

13%

INTERNET SOURCES

6%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1	Supriyono Supriyono, Ade Laelissiyamah. "Sistem Pendukung Keputusan Rekomendasi Penempatan Pegawai Menggunakan Metode Profile Matching", Jurnal Ilmiah Intech : Information Technology Journal of UMUS, 2020 Publication	3%
2	www.ijctjournal.org Internet Source	3%
3	journal.ithb.ac.id Internet Source	2%
4	repository.uin-suska.ac.id Internet Source	1%
5	jurnal.ugm.ac.id Internet Source	1%
6	studentjournal.petra.ac.id Internet Source	1%
7	journals.ums.ac.id Internet Source	1%

8

ejurnal.tunasbangsa.ac.id

Internet Source

1%

9

id.scribd.com

Internet Source

1%

10

e-jurnal.pelitanusantara.ac.id

Internet Source

1%

11

id.123dok.com

Internet Source

1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On