**PEMROGRAMAN VISUAL**

**IMPLEMENTASI DELPHI & MYSQL  
UNTUK GUEST BOOK**

TUGAS UAS



**DOSEN PENGAMPU: Wagino, M. Kom.**

Disusun Oleh :

**FAKULTAS TEKNIK INFORMATIKA**

**UNIVERSITAS ISLAM KALIMANTAN**

**MUHAMMAD ARSYAD AL BANJARI**

**BANJARMASIN**

**2025**

DAFTAR ISI

[**I.** **PEMBUATAN DATABASE** 1](#_Toc202998336)

[MEMBUAT DATABASE 1](#_Toc202998337)

[**II.** **PEMBUATAN FORM DELPHI** 3](#_Toc202998338)

[ FORM CRUD DATA 4](#_Toc202998339)

[ SOURCE CODE CRUD DATA 6](#_Toc202998340)

[ FORM QUICK REPORT 12](#_Toc202998341)

[ SOURCE CODE QUICK REPORT 14](#_Toc202998342)

[**III.** **RINGKASAN DAN KESIMPULAN** 16](#_Toc202998343)

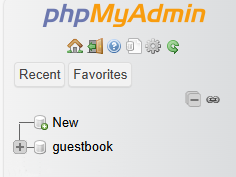
[I. Ringkasan Proyek 16](#_Toc202998344)

[II. Mekanisme Koneksi Database 16](#_Toc202998345)

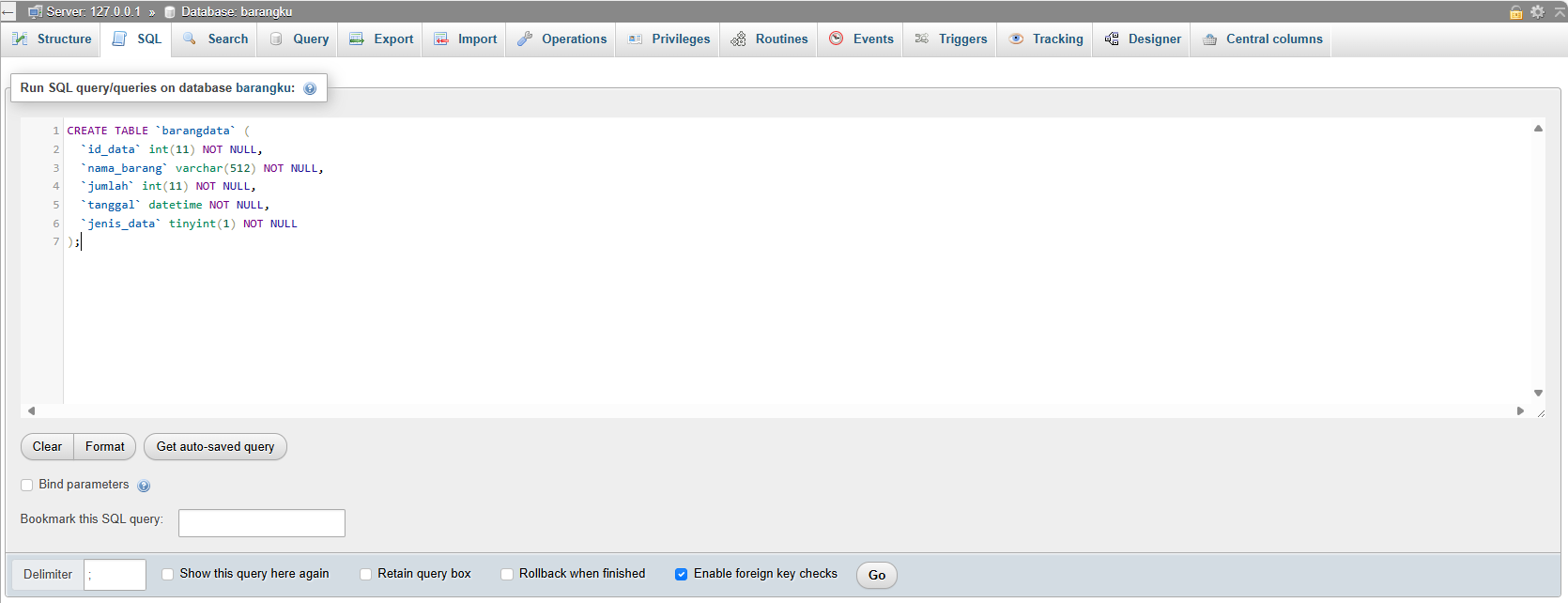
[III. Integrasi dengan QuickReport untuk Pelaporan Data 17](#_Toc202998346)

# **PEMBUATAN DATABASE**

## MEMBUAT DATABASE

Klik New > Masukan nama database “[barangku](http://localhost/phpmyadmin/index.php?route=/database/structure&db=agent_travel)” > Create

Buat tabel account dengan cara

1. Klik barangku
2. Klik SQL
3. Klik masukan Query
4. Lalu klik Go

Query untuk membuat TABLE tamu

1. CREATE TABLE `barangdata` (

2. `id\_data` int(11) NOT NULL,

3. `nama\_barang` varchar(512) NOT NULL,

4. `jumlah` int(11) NOT NULL,

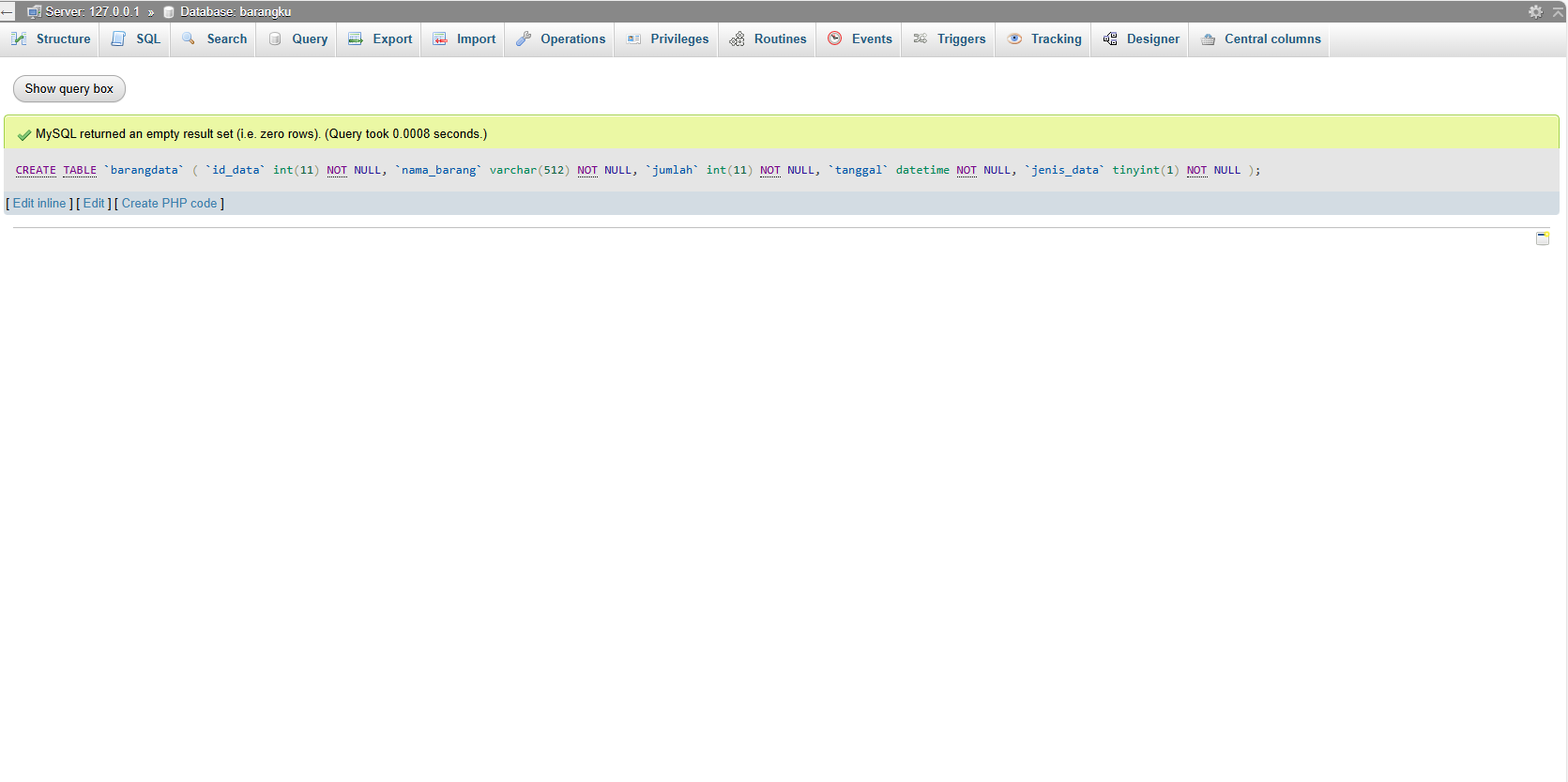
5. `tanggal` datetime NOT NULL,

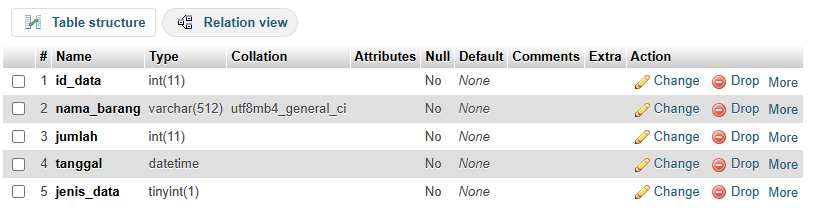
6. `jenis\_data` tinyint(1) NOT NULL

7. );

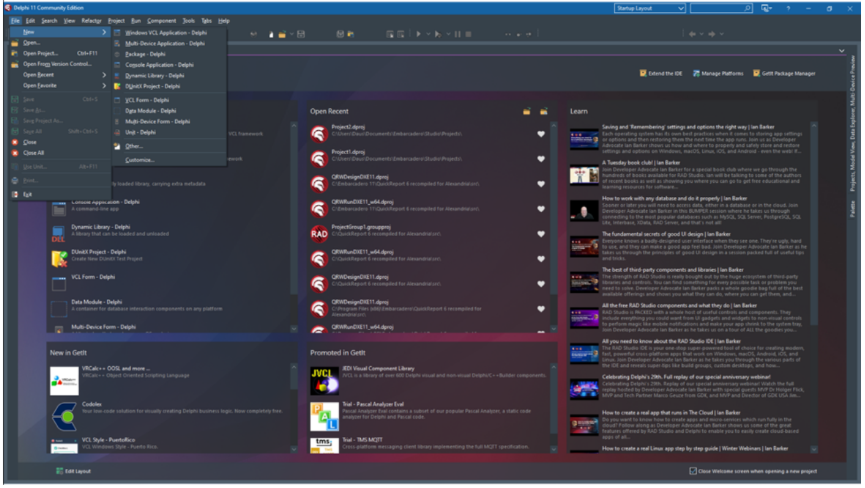
8.

Dan tampilannya akan seperti ini





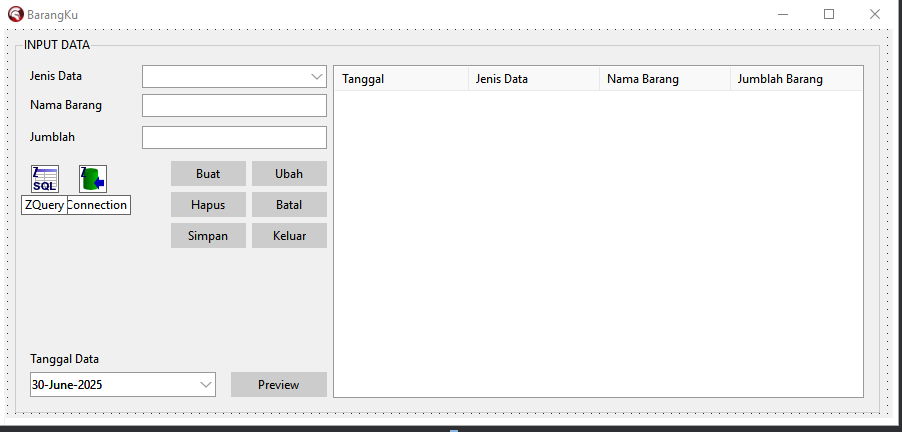
# **PEMBUATAN FORM DELPHI**

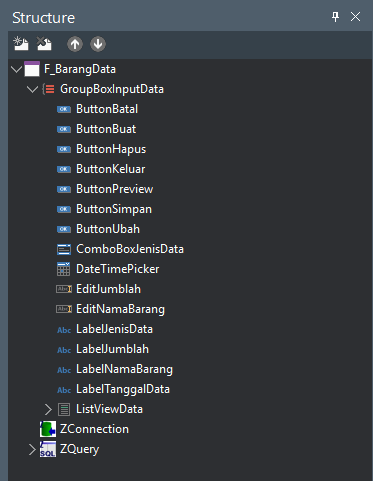
Buka Delphi > File > Windows VCL Application – Delphi

## FORM CRUD DATA

Form Name : F\_BarangData

Form Caption : BarangKu





Komponen Form F\_BarangData

|  |  |  |
| --- | --- | --- |
| **KOMPONEN** | **NAME** | **CAPTION** |
| TGroupBox | GroupBoxInputData | - |
| TButton | ButtonBatal | Batal |
| TButton | ButtonBuat | Buat |
| TButton | ButtonHapus | Hapus |
| TButton | ButtonKeluar | Keluar |
| TButton | ButtonPreview | Preview |
| TButton | ButtonSimpan | Simpan |
| TButton | ButtonUbah | Ubah |
| TComboBox | ComboBoxJenisData | - |
| TEdit | DateTimePicker | - |
| TEdit | EditJumblah | - |
| TEdit | EditNamaBarang | - |
| TLabel | LabelJenisData | JenisData |
| TLabel | LabelJumblah | Jumblah |
| TLabel | LabelNamaBarang | NamaBarang |
| TLabel | LabelTanggalData | TanggalData |
| TGroupBox | ListViewData | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |

Struktur ListViewData Column dan aktifkan semua kolom AutoSize

1. Tanggal
2. Jenis Data
3. Nama Barang
4. Jumblah Barang

## SOURCE CODE CRUD DATA

1. unit barangdata;

2.

3. interface

4.

5. uses

6. Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics, System.UITypes,

7. Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, Vcl.StdCtrls, Vcl.Menus,

8. Data.DB, ZAbstractRODataset, ZAbstractDataset, ZDataset, ZAbstractConnection,

9. ZConnection, BarangDataQuickReport;

10.

11. type

12. TF\_BarangData = class(TForm)

13. GroupBoxInputData: TGroupBox;

14. ListViewData: TListView;

15. LabelJenisData: TLabel;

16. LabelNamaBarang: TLabel;

17. ButtonBuat: TButton;

18. ButtonHapus: TButton;

19. ButtonUbah: TButton;

20. ButtonBatal: TButton;

21. ButtonKeluar: TButton;

22. DateTimePicker: TDateTimePicker;

23. LabelTanggalData: TLabel;

24. ButtonPreview: TButton;

25. ZConnection: TZConnection;

26. ZQuery: TZQuery;

27. ButtonSimpan: TButton;

28. ComboBoxJenisData: TComboBox;

29. EditJumblah: TEdit;

30. LabelJumblah: TLabel;

31. EditNamaBarang: TEdit;

32.

33. procedure FormCreate(Sender: TObject);

34.

35.

36. procedure ButtonBuatClick(Sender: TObject);

37. procedure ButtonHapusClick(Sender: TObject);

38. procedure ButtonUbahClick(Sender: TObject);

39. procedure ButtonBatalClick(Sender: TObject);

40. procedure ButtonKeluarClick(Sender: TObject);

41. procedure ButtonPreviewClick(Sender: TObject);

42. procedure ButtonSimpanClick(Sender: TObject);

43.

44.

45. procedure ListViewDataItemClick(Sender: TObject; Item: TListItem; Selected: Boolean);

46.

47. // sanitasi agar hanya angka yang dapat di input kepada EditJumblah

48. procedure EditJumblahKeyPress(Sender: TObject; var Key: Char);

49.

50. procedure ConfigureMySQLConnection;

51. private

52. { Private declarations }

53. FMode: string;

54. procedure SetStateAwal;

55. procedure SetStateEdit;

56. procedure SetStateInputBaru;

57. procedure RefreshListView;

58. procedure ClearFormFields;

59. public

60. { Public declarations }

61. end;

62.

63. var

64. F\_BarangData: TF\_BarangData;

65.

66. implementation

67.

68. {$R \*.dfm}

69.

70. procedure TF\_BarangData.EditJumblahKeyPress(Sender: TObject; var Key: Char);

71. begin

72. if not CharInSet(Key, ['0'..'9', #8]) then

73. Key := #0;

74. end;

75.

76. procedure TF\_BarangData.ClearFormFields;

77. begin

78. ComboBoxJenisData.ItemIndex := -1;

79. EditJumblah.Clear;

80. EditNamaBarang.Clear;

81. DateTimePicker.Date := Now;

82. ListViewData.Selected := nil;

83. end;

84.

85. procedure TF\_BarangData.FormCreate(Sender: TObject);

86. begin

87. ConfigureMySQLConnection;

88. DateTimePicker.Date := Now;

89. EditNamaBarang.Clear;

90. ComboBoxJenisData.Items.Clear;

91. ComboBoxJenisData.Items.Add('Barang Keluar'); // 0

92. ComboBoxJenisData.Items.Add('Barang Masuk'); // 1

93. ComboBoxJenisData.ItemIndex := -1;

94. RefreshListView;

95. ListViewData.OnSelectItem := ListViewDataItemClick;

96. ListViewData.ReadOnly := True; // Prevent editing directly in ListView

97. ListViewData.ViewStyle := vsReport;

98. ListViewData.RowSelect := True;

99. ButtonBuat.OnClick := ButtonBuatClick;

100. ButtonHapus.OnClick := ButtonHapusClick;

101. ButtonUbah.OnClick := ButtonUbahClick;

102. ButtonBatal.OnClick := ButtonBatalClick;

103. ButtonKeluar.OnClick := ButtonKeluarClick;

104. ButtonPreview.OnClick := ButtonPreviewClick;

105. ButtonSimpan.OnClick := ButtonSimpanClick;

106. EditJumblah.OnKeyPress := EditJumblahKeyPress;

107. SetStateAwal;

108. end;

109.

110. procedure TF\_BarangData.ConfigureMySQLConnection;

111. begin

112. ZConnection.HostName := 'localhost';

113. ZConnection.Database := 'barangku';

114. ZConnection.User := 'root';

115. ZConnection.Password := '';

116. ZConnection.Protocol := 'mysql';

117. ZConnection.Port := 3306;

118. ZConnection.Connect;

119. ZQuery.Connection := ZConnection;

120. if not ZConnection.Connected then

121. ShowMessage('Failed to connect to the database.');

122. end;

123.

124. procedure TF\_BarangData.SetStateAwal;

125. begin

126. ButtonBuat.Enabled := True;

127. ButtonSimpan.Enabled := False;

128. ButtonUbah.Enabled := False;

129. ButtonHapus.Enabled := False;

130. ButtonBatal.Enabled := True;

131. ButtonKeluar.Enabled := True;

132. ButtonPreview.Enabled := True;

133. ComboBoxJenisData.Enabled := False;

134. EditJumblah.Enabled := False;

135. EditNamaBarang.Enabled := False;

136. ListViewData.Enabled := True; // Enable ListView for data selection

137. ListViewData.Selected := nil; // Clear any selection

138. ClearFormFields;

139. end;

140.

141. procedure TF\_BarangData.SetStateEdit;

142. begin

143. ButtonBuat.Enabled := False;

144. ButtonSimpan.Enabled := True;

145. ButtonUbah.Enabled := True;

146. ButtonHapus.Enabled := True;

147. ButtonBatal.Enabled := True;

148. ButtonKeluar.Enabled := True;

149. ButtonPreview.Enabled := True;

150. ComboBoxJenisData.Enabled := True;

151. EditJumblah.Enabled := True;

152. EditNamaBarang.Enabled := True;

153. ListViewData.Enabled := False; // Disable ListView while inputting new data

154. end;

155.

156. procedure TF\_BarangData.SetStateInputBaru;

157. begin

158. ButtonBuat.Enabled := False;

159. ButtonSimpan.Enabled := True;

160. ButtonUbah.Enabled := False;

161. ButtonHapus.Enabled := False;

162. ButtonBatal.Enabled := True;

163. ButtonKeluar.Enabled := True;

164. ButtonPreview.Enabled := False;

165. ComboBoxJenisData.Enabled := True;

166. EditJumblah.Enabled := True;

167. EditNamaBarang.Enabled := True;

168. ListViewData.Selected := nil;

169. ListViewData.Enabled := False; // Disable ListView while inputting new data

170. ClearFormFields;

171. end;

172.

173. // REVISI

174. procedure TF\_BarangData.RefreshListView;

175. begin

176. ListViewData.Clear;

177. ZQuery.SQL.Text := 'SELECT \* FROM barangdata ORDER BY tanggal DESC';

178. ZQuery.Open;

179. while not ZQuery.Eof do

180. begin

181. with ListViewData.Items.Add do

182. begin

183. Caption := FormatDateTime('d mmmm yyyy hh:nn:ss', ZQuery.FieldByName('tanggal').AsDateTime);

184. if ZQuery.FieldByName('jenis\_data').AsInteger = 0 then

185. SubItems.Add('Barang Keluar')

186. else

187. SubItems.Add('Barang Masuk');

188. SubItems.Add(ZQuery.FieldByName('nama\_barang').AsString);

189. SubItems.Add(ZQuery.FieldByName('jumlah').AsString);

190. Data := Pointer(ZQuery.FieldByName('id\_data').AsInteger);

191. end;

192. ZQuery.Next;

193. end;

194. ZQuery.Close;

195. end;

196.

197. procedure TF\_BarangData.ButtonBuatClick(Sender: TObject);

198. begin

199. FMode := 'create';

200. SetStateInputBaru;

201. end;

202.

203. procedure TF\_BarangData.ButtonSimpanClick(Sender: TObject);

204. var

205. jenisTransaksi: Integer;

206. begin

207. if (ComboBoxJenisData.ItemIndex = -1) or

208. (EditJumblah.Text = '') or

209. (EditNamaBarang.Text = '') then

210. begin

211. ShowMessage('Silakan lengkapi semua data yang wajib diisi.');

212. Exit;

213. end;

214. if not ZConnection.Connected then

215. begin

216. ShowMessage('Koneksi ke database belum berhasil.');

217. Exit;

218. end;

219. jenisTransaksi := ComboBoxJenisData.ItemIndex; // 0: Barang Keluar, 1: Barang Masuk

220. try

221. if FMode = 'create' then

222. begin

223. ZQuery.SQL.Text := 'INSERT INTO barangdata (tanggal, jenis\_data, nama\_barang, jumlah) VALUES (NOW(), :jenis\_data, :nama\_barang, :jumlah)';

224. ZQuery.ParamByName('jenis\_data').AsInteger := jenisTransaksi;

225. ZQuery.ParamByName('nama\_barang').AsString := EditNamaBarang.Text;

226. ZQuery.ParamByName('jumlah').AsString := EditJumblah.Text;

227. ZQuery.ExecSQL;

228. ShowMessage('Data keuangan berhasil ditambahkan.');

229. end

230. else if (FMode = 'edit') and (ListViewData.Selected <> nil) then

231. begin

232. ZQuery.SQL.Text := 'UPDATE barangdata SET jenis\_data = :jenis\_data, nama\_barang = :nama\_barang, jumlah = :jumlah WHERE id\_data = :id\_data';

233. ZQuery.ParamByName('jenis\_data').AsInteger := jenisTransaksi;

234. ZQuery.ParamByName('nama\_barang').AsString := EditNamaBarang.Text;

235. ZQuery.ParamByName('jumlah').AsString := EditJumblah.Text;

236. ZQuery.ParamByName('id\_data').AsInteger := Integer(ListViewData.Selected.Data);

237. ZQuery.ExecSQL;

238. ShowMessage('Data keuangan berhasil diubah.');

239. end;

240. RefreshListView;

241. SetStateAwal;

242. FMode := '';

243. except

244. on E: Exception do

245. ShowMessage('Gagal menyimpan data keuangan: ' + E.Message);

246. end;

247. end;

248.

249. procedure TF\_BarangData.ButtonHapusClick(Sender: TObject);

250. begin

251.

252. if ListViewData.Selected = nil then

253. begin

254. ShowMessage('Silakan pilih data keuangan yang ingin dihapus.');

255. Exit;

256. end;

257.

258. if not ZConnection.Connected then

259. begin

260. ShowMessage('Koneksi ke database belum berhasil.');

261. Exit;

262. end;

263.

264. if MessageDlg('Apakah Anda yakin ingin menghapus data ini?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then

265. begin

266. try

267. ZQuery.SQL.Text := 'DELETE FROM barangdata WHERE id\_data = :id';

268. ZQuery.ParamByName('id').AsInteger := Integer(ListViewData.Selected.Data);

269. ZQuery.ExecSQL;

270. ShowMessage('Data keuangan berhasil dihapus.');

271. RefreshListView;

272. SetStateAwal;

273. except

274. on E: Exception do

275. ShowMessage('Gagal menghapus data keuangan: ' + E.Message);

276. end;

277. end;

278. end;

279.

280. procedure TF\_BarangData.ButtonUbahClick(Sender: TObject);

281. begin

282. if ListViewData.Selected = nil then

283. begin

284. ShowMessage('Silakan pilih data keuangan yang ingin diubah.');

285. Exit;

286. end;

287. FMode := 'edit';

288. SetStateEdit;

289. end;

290.

291. procedure TF\_BarangData.ButtonPreviewClick(Sender: TObject);

292. var

293. LFinanceReport: TF\_BarangkuQuickReport;

294. begin

295. LFinanceReport := TF\_BarangkuQuickReport.Create(Self);

296. try

297. LFinanceReport.GenerateReport(DateTimePicker.Date);

298. finally

299. LFinanceReport.Free;

300. end;

301. end;

302.

303. procedure TF\_BarangData.ListViewDataItemClick(Sender: TObject; Item: TListItem; Selected: Boolean);

304. begin

305.

306. if Selected then

307. begin

308. if Item.SubItems[0] = 'Barang Keluar' then

309. ComboBoxJenisData.ItemIndex := 0

310. else if Item.SubItems[0] = 'Barang Masuk' then

311. ComboBoxJenisData.ItemIndex := 1

312. else

313. ComboBoxJenisData.ItemIndex := -1;

314. EditNamaBarang.Text := Item.SubItems[1];

315. EditJumblah.Text := Item.SubItems[2];

316. ButtonUbah.Enabled := True;

317. ButtonHapus.Enabled := True;

318. end

319. else

320. begin

321. ButtonUbah.Enabled := False;

322. ButtonHapus.Enabled := False;

323. end;

324. end;

325.

326. procedure TF\_BarangData.ButtonBatalClick(Sender: TObject);

327. begin

328. ClearFormFields;

329. SetStateAwal;

330. end;

331.

332. procedure TF\_BarangData.ButtonKeluarClick(Sender: TObject);

333. begin

334. Application.Terminate;

335. end;

336.

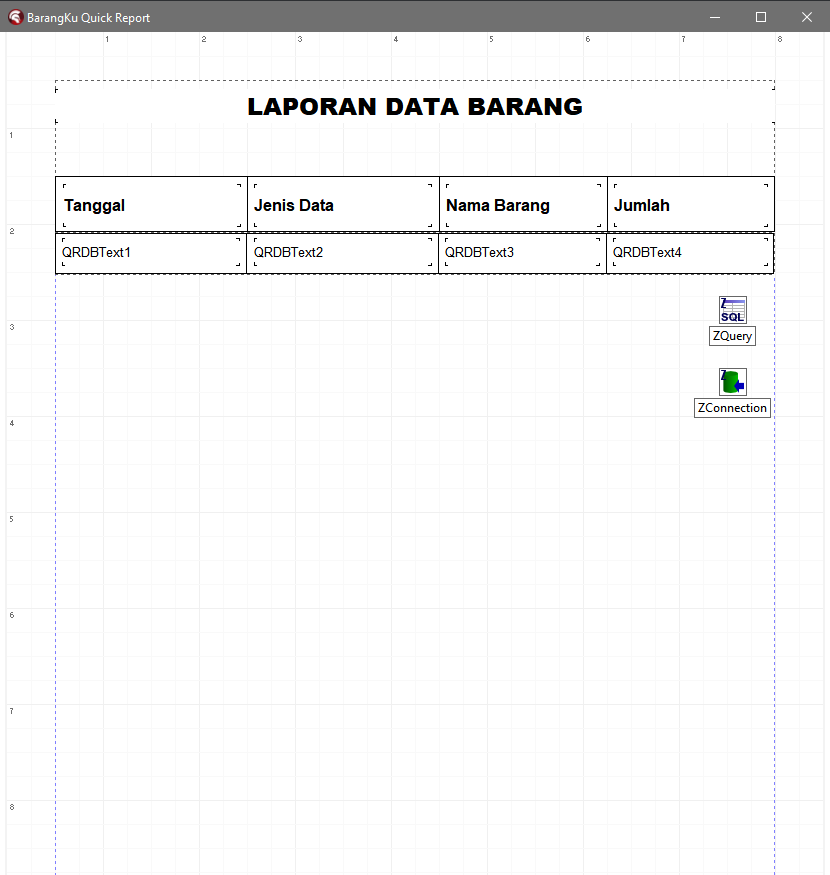
337. end.

338.

## FORM QUICK REPORT

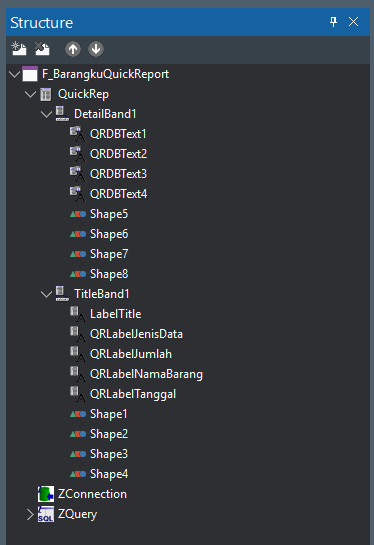
Form Name : F\_BarangkuQuickReport

Form Caption : BarangKu Quick Report



Komponen Form F\_GuestDataQuickReport

|  |  |  |
| --- | --- | --- |
| **KOMPONEN** | **NAME** | **CAPTION** |
| T QuickRep | QuickRep | - |
| TQRBand | DetailBand1 | - |
| TQRDBText | QRDBText1 | - |
| TQRDBText | QRDBText2 | - |
| TQRDBText | QRDBText3 | - |
| TQRDBText | QRDBText4 | - |
| TShape | Shape5 | - |
| TShape | Shapeб | - |
| TShape | Shape7 | - |
| TShape | Shape8 | - |
| TQRBand | TitleBand1 | - |
| TLabel | LabelTitle | LAPORAN DATA BARANG |
| TQRLabel | QRLabelTanggal | Tanggal |
| TQRLabel | QRLabelJenisData | JenisData |
| TQRLabel | QRLabelNamaBarang | NamaBarang |
| TQRLabel | QRLabelJumlah | Jumlah |
| TShape | Shape1 | - |
| TShape | Shape2 | - |
| TShape | Shape3 | - |
| TShape | Shape4 | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |



## SOURCE CODE QUICK REPORT

1. unit BarangDataQuickReport;

2.

3. interface

4.

5. uses

6. Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

7. Vcl.Controls, Vcl.Forms, Vcl.Dialogs, QuickRpt, QRCtrls, Vcl.ExtCtrls,

8. ZAbstractConnection, ZConnection, Data.DB, ZAbstractRODataset,

9. ZAbstractDataset, ZDataset;

10.

11. type

12. TF\_BarangkuQuickReport = class(TForm)

13. QuickRep: TQuickRep;

14. TitleBand1: TQRBand;

15. DetailBand1: TQRBand;

16. LabelTitle: TQRLabel;

17. Shape1: TShape;

18. Shape2: TShape;

19. Shape3: TShape;

20. Shape4: TShape;

21. QRLabelTanggal: TQRLabel;

22. QRLabelJenisData: TQRLabel;

23. QRLabelNamaBarang: TQRLabel;

24. QRLabelJumlah: TQRLabel;

25. Shape5: TShape;

26. Shape6: TShape;

27. Shape7: TShape;

28. Shape8: TShape;

29. QRDBText1: TQRDBText;

30. QRDBText2: TQRDBText;

31. QRDBText3: TQRDBText;

32. QRDBText4: TQRDBText;

33. ZQuery: TZQuery;

34. ZConnection: TZConnection;

35. private

36. { Private declarations }

37. procedure ConnectionSetup;

38. procedure QuerySetup(ATanggal: TDateTime);

39. procedure DetailBandSetup;

40.

41. public

42. { Public declarations }

43. procedure GenerateReport(ATanggal: TDateTime);

44.

45.

46. end;

47.

48. var

49. F\_BarangkuQuickReport: TF\_BarangkuQuickReport;

50.

51. implementation

52.

53. {$R \*.dfm}

54.

55. procedure TF\_BarangkuQuickReport.ConnectionSetup;

56. begin

57. ZConnection.HostName := 'localhost';

58. ZConnection.Database := 'barangku';

59. ZConnection.User := 'root';

60. ZConnection.Password := '';

61. ZConnection.Protocol := 'mysql';

62. ZConnection.Port := 3306;

63. ZConnection.Connect;

64. ZQuery.Connection := ZConnection;

65. QuickRep.DataSet := ZQuery;

66. end;

67.

68. procedure TF\_BarangkuQuickReport.QuerySetup(ATanggal: TDateTime);

69. begin

70. ZQuery.SQL.Text :=

71. 'SELECT tanggal, ' +

72. 'CASE WHEN jenis\_data = 0 THEN ''Barang Keluar'' ELSE ''Barang Masuk'' END AS jenis\_data\_str, ' +

73. 'jumlah, nama\_barang ' +

74. 'FROM barangdata WHERE DATE(tanggal) = :tanggal';

75. ZQuery.ParamByName('tanggal').AsDate := ATanggal;

76. ZQuery.Open;

77. end;

78.

79. procedure TF\_BarangkuQuickReport.GenerateReport(ATanggal: TDateTime);

80. begin

81. ConnectionSetup;

82. DetailBandSetup;

83. QuerySetup(ATanggal);

84.

85. if ZQuery.IsEmpty then

86. begin

87. ShowMessage('Tidak ada data untuk tanggal yang dipilih.');

88. Exit;

89. end;

90.

91. QuickRep.Preview;

92. end;

93.

94. procedure TF\_BarangkuQuickReport.DetailBandSetup;

95. begin

96. QRDBText1.DataSet := ZQuery;

97. QRDBText2.DataSet := ZQuery;

98. QRDBText3.DataSet := ZQuery;

99. QRDBText4.DataSet := ZQuery;

100. QRDBText1.DataField := 'tanggal';

101. QRDBText2.DataField := 'jenis\_data\_str';

102. QRDBText3.DataField := 'nama\_barang';

103. QRDBText4.DataField := 'jumlah';

104. QRDBText1.Mask := 'd mmmm yyyy hh:nn:ss';

105. end;

106.

107. end.

108.

# **RINGKASAN DAN KESIMPULAN**

## **Ringkasan Proyek**

Proyek ini bertujuan untuk mengembangkan sebuah aplikasi pengelolaan data barang, bernama "BarangKu". Pengembangan aplikasi ini memanfaatkan Delphi sebagai lingkungan pengembangan antarmuka pengguna dan MySQL sebagai sistem manajemen basis data. Aplikasi "BarangKu" dirancang untuk memfasilitasi pencatatan data barang, meliputi jenis data (masuk/keluar), nama barang, jumlah, dan tanggal transaksi. Fitur-fitur utama yang diimplementasikan dalam aplikasi ini mencakup operasi dasar manajemen data (Create, Read, Update, Delete - CRUD) serta kapabilitas untuk menghasilkan laporan data barang menggunakan QuickReport. Pemanfaatan Delphi memungkinkan pengembangan antarmuka yang intuitif dan fungsional, sementara MySQL menyediakan solusi basis data yang andal dan efisien untuk penyimpanan dan pengelolaan data. Integrasi kedua teknologi ini memastikan aplikasi dapat berinteraksi dengan data secara dinamis dan responsif.

## II. Mekanisme Koneksi Database

Koneksi antara aplikasi Delphi dan database MySQL merupakan aspek krusial yang memungkinkan aplikasi untuk melakukan operasi pembacaan, penulisan, modifikasi, dan penghapusan data. Dalam proyek ini, koneksi database diatur melalui komponen TZConnection dan TZQuery yang merupakan bagian dari ZeosLib, sebuah pustaka yang menyediakan antarmuka fleksibel untuk berbagai jenis database, termasuk MySQL. Prosedur koneksi database melibatkan langkah-langkah berikut:

* **Inisialisasi Komponen Koneksi**: Komponen TZConnection (misalnya, ZConnection) digunakan untuk mendefinisikan parameter-parameter koneksi ke database MySQL. Parameter-parameter ini dikonfigurasi secara programatis dalam prosedur

ConfigureMySQLConnection, yang dieksekusi pada saat form utama aplikasi dibuat (FormCreate).

* **Konfigurasi Parameter Koneksi**: Prosedur ConfigureMySQLConnection mengatur properti-properti esensial dari TZConnection sebagai berikut:
  + ZConnection.HostName := 'localhost';: Menentukan alamat host server database. Dalam konteks ini, 'localhost' mengindikasikan bahwa database beroperasi pada mesin yang sama dengan aplikasi.
  + ZConnection.Database := 'barangku';: Menentukan nama database yang akan diakses, yaitu 'barangku'.
  + ZConnection.User := 'root';: Menentukan nama pengguna database. 'root' umumnya merupakan pengguna default dengan hak akses penuh pada sebagian besar instalasi MySQL lokal.
  + ZConnection.Password := '';: Menentukan kata sandi untuk pengguna database. Dalam implementasi ini, kata sandi dibiarkan kosong. Perlu ditekankan bahwa untuk lingkungan produksi, penggunaan kata sandi kosong sangat tidak direkomendasikan demi menjaga integritas dan keamanan data.
  + ZConnection.Protocol := 'mysql';: Menentukan protokol koneksi yang digunakan, yaitu MySQL.
  + ZConnection.Port := 3306;: Menentukan nomor port yang digunakan oleh server MySQL, yang secara standar adalah 3306.
* **Pembentukan Koneksi**: Setelah seluruh parameter dikonfigurasi, perintah ZConnection.Connect; dieksekusi untuk membangun koneksi fisik ke database MySQL. Apabila koneksi gagal terjalin, sebuah pesan kesalahan akan ditampilkan kepada pengguna.
* **Asosiasi Query dengan Koneksi**: Komponen TZQuery (misalnya, ZQuery) berfungsi untuk mengeksekusi perintah SQL. Penting untuk mengasosiasikan

TZQuery dengan TZConnection yang telah terhubung, dengan menetapkan properti Connection pada objek ZQuery ke instans ZConnection yang telah terhubung. Hal ini memastikan bahwa setiap perintah SQL yang dieksekusi oleh

ZQuery akan diarahkan ke database yang benar.

Melalui pengaturan koneksi ini, aplikasi dapat berinteraksi secara mulus dengan database MySQL, memfasilitasi operasi CRUD dan pengambilan data untuk keperluan pelaporan.

## Integrasi dengan QuickReport untuk Pelaporan Data

Untuk menghasilkan laporan yang terstruktur dan akurat menggunakan QuickReport, integrasi yang tepat antara komponen data (yaitu TZQuery) dan komponen visual QuickReport sangat penting. Proses ini memastikan bahwa data yang diambil dari database dapat disajikan dengan presisi pada laporan.

Langkah-langkah integrasi QuickReport adalah sebagai berikut:

* **Penghubungan QuickRep dengan Sumber Data**: Komponen TQuickRep (misalnya, QuickRep) berfungsi sebagai wadah utama untuk laporan. Agar

TQuickRep dapat mengakses data, properti DataSet-nya harus dihubungkan ke komponen TZQuery yang telah mengambil data dari database. Ini diimplementasikan dengan baris kode:

QuickRep.DataSet := ZQuery;. Dengan konfigurasi ini, TQuickRep dapat mengidentifikasi sumber data yang akan digunakan untuk presentasi laporan.

* **Konfigurasi Query untuk Pelaporan**: Sebelum laporan ditampilkan, TZQuery perlu dieksekusi dengan perintah SQL yang spesifik sesuai dengan data yang akan dilaporkan. Dalam prosedur

QuerySetup(ATanggal: TDateTime), properti ZQuery.SQL.Text diatur untuk memilih kolom tanggal, jenis\_data\_str, jumlah, dan nama\_barang dari tabel barangdata, dengan kriteria filter berdasarkan tanggal (WHERE DATE(tanggal) = :tanggal). Parameter :tanggal kemudian diisi dengan nilai ATanggal yang diberikan. Selanjutnya, ZQuery.Open; dieksekusi untuk membuka dataset dan mengambil data yang relevan.

* **Pemetaan Data ke Komponen TQRDBText**: Komponen TQRDBText digunakan untuk menampilkan nilai dari kolom-kolom database pada laporan. Setiap

TQRDBText harus dihubungkan ke DataSet yang sesuai dan DataField yang spesifik. Dalam prosedur

DetailBandSetup, setiap TQRDBText dikonfigurasi sebagai berikut:

* + QRDBText1.DataSet := ZQuery;
  + QRDBText1.DataField := 'tanggal';

Konfigurasi serupa diterapkan untuk

QRDBText2 (jenis data), QRDBText3 (nama barang), dan QRDBText4 (jumlah). Pengaturan ini memastikan bahwa

QRDBText1 akan menampilkan data dari kolom tanggal, QRDBText2 dari jenis\_data\_str, dan seterusnya, sesuai dengan pemetaan yang telah ditetapkan.

* **Pemformatan Data (Masking)**: Untuk data yang memerlukan format presentasi khusus, seperti tanggal dan waktu, properti Mask pada TQRDBText dapat digunakan. Sebagai contoh,

QRDBText1.Mask := 'd mmmm yyyy hh:nn:ss'; memastikan bahwa kolom tanggal akan ditampilkan dalam format tanggal dan waktu yang terstandardisasi (hari bulan tahun jam:menit:detik).

Dengan implementasi langkah-langkah ini, QuickReport dapat secara dinamis mengambil data dari

TZQuery dan menyajikannya dengan rapi pada laporan, memastikan bahwa setiap komponen TQRDBText menampilkan data yang akurat sesuai dengan pemetaan kolom yang telah didefinisikan