**PEMROGRAMAN VISUAL**

**IMPLEMENTASI DELPHI & MYSQL  
UNTUK BAYARHUTANG**

TUGAS UAS



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

Disusun Oleh :

Muhammad Rizki 2410010144

**FAKULTAS TEKNIK INFORMATIKA**

**UNIVERSITAS ISLAM KALIMANTAN**

**MUHAMMAD ARSYAD AL BANJARI**

**BANJARMASIN**

**2025**

DAFTAR ISI

[**I.** **PEMBUATAN DATABASE** 1](#_Toc203169645)

[ MEMBUAT DATABASE 1](#_Toc203169646)

[**II.** **PEMBUATAN FORM DELPHI** 3](#_Toc203169647)

[ FORM CRUD DATA 4](#_Toc203169648)

[ SOURCE CODE CRUD DATA 6](#_Toc203169649)

[ FORM QUICK REPORT 12](#_Toc203169650)

[ SOURCE CODE QUICK REPORT 14](#_Toc203169651)

[**III.** **RINGKASAN DAN KESIMPULAN** 17](#_Toc203169652)

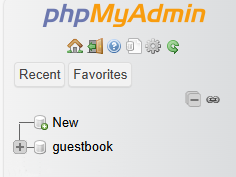
[1. Ringkasan Proyek 17](#_Toc203169653)

[2. Mekanisme Koneksi Database 17](#_Toc203169654)

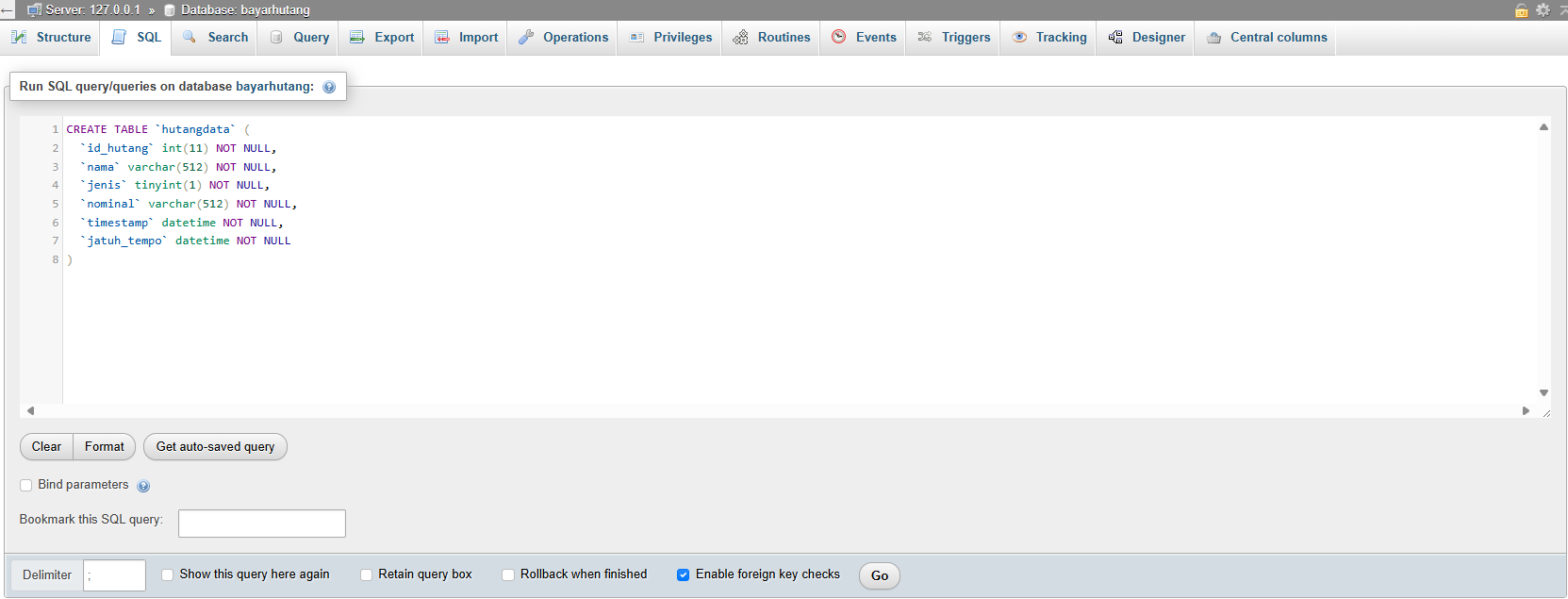
[3. Integrasi dengan QuickReport untuk Pelaporan Data 18](#_Toc203169655)

# **PEMBUATAN DATABASE**

## MEMBUAT DATABASE

Klik New > Masukan nama database “[smartclass](http://localhost/phpmyadmin/index.php?route=/database/structure&db=agent_travel" \o "Jump to database 'agent_travel')” > Create

Buat tabel account dengan cara

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

Query untuk membuat TABLE absensi

1. CREATE TABLE `hutangdata` (

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

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

4. `jenis` tinyint(1) NOT NULL,

5. `nominal` varchar(512) NOT NULL,

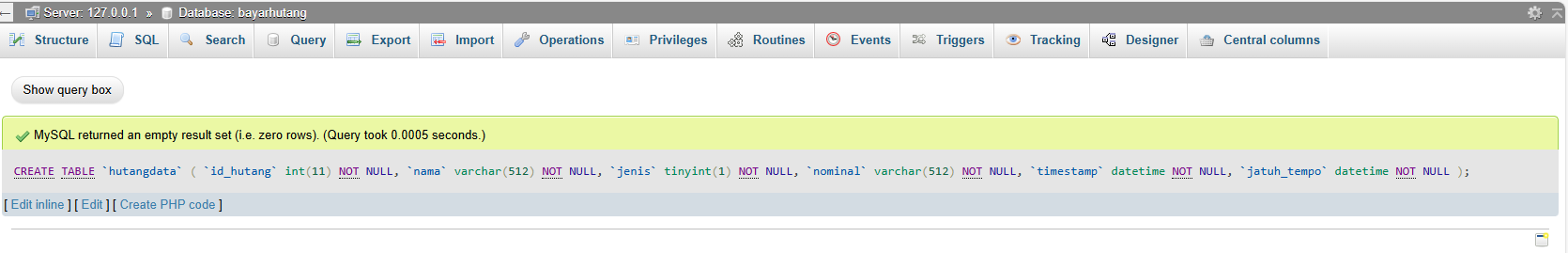
6. `timestamp` datetime NOT NULL,

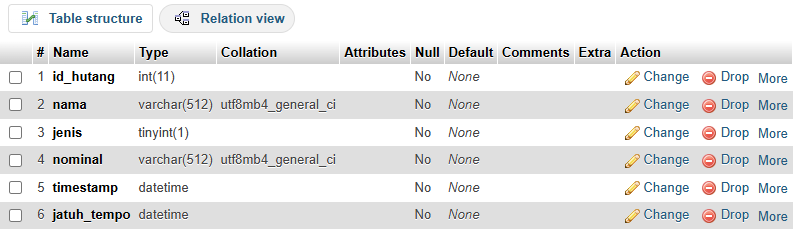
7. `jatuh\_tempo` datetime NOT NULL

8. );

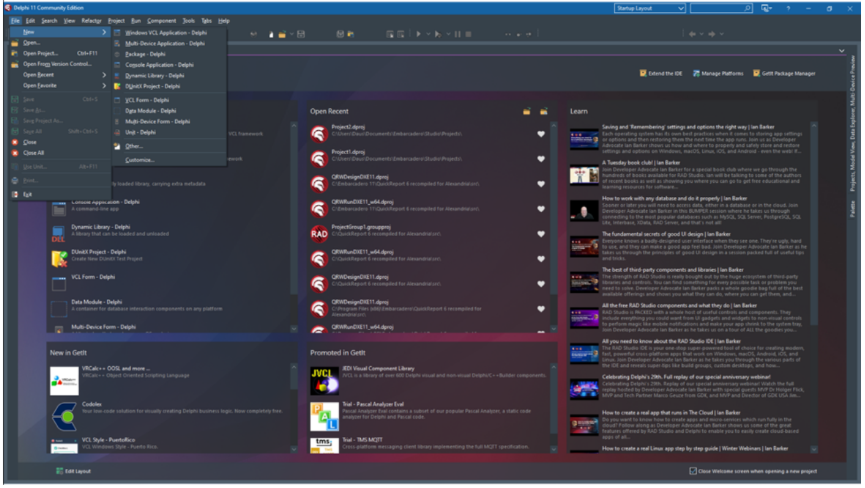
9.

Dan tampilannya akan seperti ini





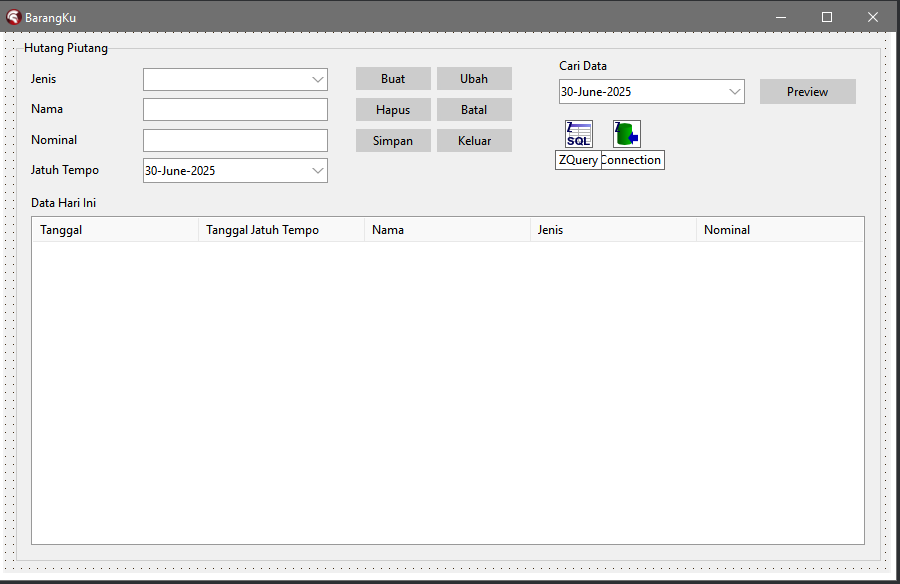
# **PEMBUATAN FORM DELPHI**

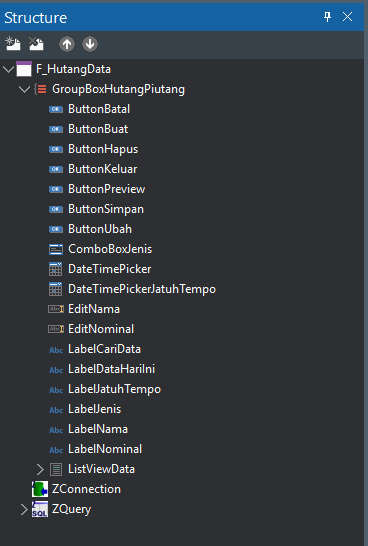
Buka Delphi > File > Windows VCL Application – Delphi

## FORM CRUD DATA

Form Name F\_HutangData

Form Caption : Bayar Hutang





Komponen Form : F\_HutangData

|  |  |  |
| --- | --- | --- |
| **KOMPONEN** | **NAME** | **CAPTION** |
| TGroupBox | GroupBoxHutangPiutang | Hutang Piutang |
| TButton | ButtonBatal | Batal |
| TButton | ButtonBuat | Buat |
| TButton | ButtonHapus | Hapus |
| TButton | ButtonKeluar | Keluar |
| TButton | ButtonPreview | Preview |
| TButton | ButtonSimpan | Simpan |
| TButton | ButtonUbah | Ubah |
| TButton | ComboBoxJenis | - |
| TButton | DateTimePicker | - |
| TComboBox | DateTimePickerJatuhTempo | - |
| TDateTimePicker | EditNama | - |
| TDateTimePicker | EditNominal | - |
| TEdit | LabelCariData | Cari Data |
| TEdit | LabelDataHariIni | Data Hari Ini |
| TEdit | LabelJatuhTempo | Jatuh Tempo |
| TLabel | LabelJenis | Jenis |
| TLabel | LabelNama | Nama |
| TLabel | LabelNominal | Nominal |
| TLabel | ListViewData | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |

Struktur ListViewData Column dan aktifkan semua kolom AutoSize

1. Tanggal
2. Tanggal Jatuh Tempo
3. Nama
4. Jenis
5. Nominal

## SOURCE CODE CRUD DATA

1. unit hutangdata;

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, HutangDataQuickReport;

10.

11. type

12. TF\_HutangData = class(TForm)

13. GroupBoxHutangPiutang: TGroupBox;

14. ListViewData: TListView;

15. LabelJenis: TLabel;

16. LabelNama: TLabel;

17. ButtonBuat: TButton;

18. ButtonHapus: TButton;

19. ButtonUbah: TButton;

20. ButtonBatal: TButton;

21. ButtonKeluar: TButton;

22. DateTimePicker: TDateTimePicker;

23. LabelCariData: TLabel;

24. ButtonPreview: TButton;

25. ZConnection: TZConnection;

26. ZQuery: TZQuery;

27. ButtonSimpan: TButton;

28. ComboBoxJenis: TComboBox;

29. EditNominal: TEdit;

30. LabelNominal: TLabel;

31. EditNama: TEdit;

32. LabelJatuhTempo: TLabel;

33. DateTimePickerJatuhTempo: TDateTimePicker;

34. LabelDataHariIni: TLabel;

35.

36. procedure FormCreate(Sender: TObject);

37.

38.

39. procedure ButtonBuatClick(Sender: TObject);

40. procedure ButtonHapusClick(Sender: TObject);

41. procedure ButtonUbahClick(Sender: TObject);

42. procedure ButtonBatalClick(Sender: TObject);

43. procedure ButtonKeluarClick(Sender: TObject);

44. procedure ButtonPreviewClick(Sender: TObject);

45. procedure ButtonSimpanClick(Sender: TObject);

46.

47.

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

49.

50. // sanitasi agar hanya angka yang dapat di input kepada EditNominal

51. procedure EditNominalKeyPress(Sender: TObject; var Key: Char);

52.

53. procedure ConfigureMySQLConnection;

54. private

55. { Private declarations }

56. FMode: string;

57. procedure SetStateAwal;

58. procedure SetStateEdit;

59. procedure SetStateInputBaru;

60. procedure RefreshListView;

61. procedure ClearFormFields;

62. function RupiahFormat(const Value: string): string;

63. public

64. { Public declarations }

65. end;

66.

67.

68. var

69. F\_HutangData: TF\_HutangData;

70.

71. implementation

72.

73. {$R \*.dfm}

74.

75. function TF\_HutangData.RupiahFormat(const Value: string): string;

76. begin

77. // Format as Rupiah, no decimals, no comma, with Rp. prefix

78. try

79. Result := 'Rp. ' + FormatFloat('#,0', StrToFloat(Value));

80. except

81. Result := 'Rp. ' + Value;

82. end;

83. end;

84.

85.

86. procedure TF\_HutangData.EditNominalKeyPress(Sender: TObject; var Key: Char);

87. begin

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

89. Key := #0;

90. end;

91.

92. procedure TF\_HutangData.ClearFormFields;

93. begin

94. ComboBoxJenis.ItemIndex := -1;

95. EditNominal.Clear;

96. EditNama.Clear;

97. DateTimePicker.Date := Now;

98. ListViewData.Selected := nil;

99. end;

100.

101. procedure TF\_HutangData.FormCreate(Sender: TObject);

102. begin

103. ConfigureMySQLConnection;

104. DateTimePicker.Date := Now;

105. EditNama.Clear;

106. ComboBoxJenis.Items.Clear;

107. ComboBoxJenis.Items.Add('Hutang'); // 0

108. ComboBoxJenis.Items.Add('Piutang'); // 1

109. ComboBoxJenis.ItemIndex := -1;

110. RefreshListView;

111. ListViewData.OnSelectItem := ListViewDataItemClick;

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

113. ListViewData.ViewStyle := vsReport;

114. ListViewData.RowSelect := True;

115. ButtonBuat.OnClick := ButtonBuatClick;

116. ButtonHapus.OnClick := ButtonHapusClick;

117. ButtonUbah.OnClick := ButtonUbahClick;

118. ButtonBatal.OnClick := ButtonBatalClick;

119. ButtonKeluar.OnClick := ButtonKeluarClick;

120. ButtonPreview.OnClick := ButtonPreviewClick;

121. ButtonSimpan.OnClick := ButtonSimpanClick;

122. EditNominal.OnKeyPress := EditNominalKeyPress;

123. SetStateAwal;

124. end;

125.

126. procedure TF\_HutangData.ConfigureMySQLConnection;

127. begin

128. ZConnection.HostName := 'localhost';

129. ZConnection.Database := 'bayarhutang';

130. ZConnection.User := 'root';

131. ZConnection.Password := '';

132. ZConnection.Protocol := 'mysql';

133. ZConnection.Port := 3306;

134. ZConnection.Connect;

135. ZQuery.Connection := ZConnection;

136. if not ZConnection.Connected then

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

138. end;

139.

140. procedure TF\_HutangData.SetStateAwal;

141. begin

142. ButtonBuat.Enabled := True;

143. ButtonSimpan.Enabled := False;

144. ButtonUbah.Enabled := False;

145. ButtonHapus.Enabled := False;

146. ButtonBatal.Enabled := True;

147. ButtonKeluar.Enabled := True;

148. ButtonPreview.Enabled := True;

149. ComboBoxJenis.Enabled := False;

150. EditNominal.Enabled := False;

151. EditNama.Enabled := False;

152. DateTimePickerJatuhTempo.Enabled := False;

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

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

155. ClearFormFields;

156. end;

157.

158. procedure TF\_HutangData.SetStateEdit;

159. begin

160. ButtonBuat.Enabled := False;

161. ButtonSimpan.Enabled := True;

162. ButtonUbah.Enabled := True;

163. ButtonHapus.Enabled := True;

164. ButtonBatal.Enabled := True;

165. ButtonKeluar.Enabled := True;

166. ButtonPreview.Enabled := True;

167. ComboBoxJenis.Enabled := True;

168. EditNominal.Enabled := True;

169. EditNama.Enabled := True;

170. DateTimePickerJatuhTempo.Enabled := True;

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

172. end;

173.

174. procedure TF\_HutangData.SetStateInputBaru;

175. begin

176. ButtonBuat.Enabled := False;

177. ButtonSimpan.Enabled := True;

178. ButtonUbah.Enabled := False;

179. ButtonHapus.Enabled := False;

180. ButtonBatal.Enabled := True;

181. ButtonKeluar.Enabled := True;

182. ButtonPreview.Enabled := False;

183. ComboBoxJenis.Enabled := True;

184. EditNominal.Enabled := True;

185. EditNama.Enabled := True;

186. DateTimePickerJatuhTempo.Enabled := True;

187. ListViewData.Selected := nil;

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

189. ClearFormFields;

190. end;

191.

192. // REVISI

193. procedure TF\_HutangData.RefreshListView;

194. begin

195. ListViewData.Clear;

196. ZQuery.SQL.Text := 'SELECT \* FROM hutangdata ORDER BY timestamp DESC';

197. ZQuery.Open;

198. while not ZQuery.Eof do

199. begin

200. with ListViewData.Items.Add do

201. begin

202. Caption := FormatDateTime('d mmmm yyyy', ZQuery.FieldByName('timestamp').AsDateTime);

203. SubItems.Add(FormatDateTime('d mmmm yyyy', ZQuery.FieldByName('jatuh\_tempo').AsDateTime));

204. SubItems.Add(ZQuery.FieldByName('nama').AsString);

205. if ZQuery.FieldByName('jenis').AsInteger = 0 then

206. SubItems.Add('Hutang')

207. else

208. SubItems.Add('Piutang');

209. SubItems.Add(RupiahFormat(ZQuery.FieldByName('nominal').AsString));

210. Data := Pointer(ZQuery.FieldByName('id\_hutang').AsInteger);

211. end;

212. ZQuery.Next;

213. end;

214. ZQuery.Close;

215. end;

216.

217. procedure TF\_HutangData.ButtonBuatClick(Sender: TObject);

218. begin

219. FMode := 'create';

220. SetStateInputBaru;

221. end;

222.

223. procedure TF\_HutangData.ButtonSimpanClick(Sender: TObject);

224. var

225. jenisTransaksi: Integer;

226. begin

227. if (ComboBoxJenis.ItemIndex = -1) or

228. (EditNominal.Text = '') or

229. (EditNama.Text = '') then

230. begin

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

232. Exit;

233. end;

234. if not ZConnection.Connected then

235. begin

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

237. Exit;

238. end;

239. jenisTransaksi := ComboBoxJenis.ItemIndex; // 0: Hutang, 1: Piutang

240. try

241. if FMode = 'create' then

242. begin

243. ZQuery.SQL.Text := 'INSERT INTO hutangdata (timestamp, jatuh\_tempo, jenis, nama, nominal) VALUES (NOW(), :jatuh\_tempo, :jenis, :nama, :nominal)';

244. ZQuery.ParamByName('jatuh\_tempo').AsDateTime := DateTimePickerJatuhTempo.DateTime;

245. ZQuery.ParamByName('jenis').AsInteger := jenisTransaksi;

246. ZQuery.ParamByName('nama').AsString := EditNama.Text;

247. ZQuery.ParamByName('nominal').AsString := EditNominal.Text;

248. ZQuery.ExecSQL;

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

250. end

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

252. begin

253. ZQuery.SQL.Text := 'UPDATE hutangdata SET jenis = :jenis, nama = :nama, nominal = :nominal WHERE id\_hutang = :id\_hutang';

254. ZQuery.ParamByName('jenis').AsInteger := jenisTransaksi;

255. ZQuery.ParamByName('nama').AsString := EditNama.Text;

256. ZQuery.ParamByName('nominal').AsString := EditNominal.Text;

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

258. ZQuery.ExecSQL;

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

260. end;

261. RefreshListView;

262. SetStateAwal;

263. FMode := '';

264. except

265. on E: Exception do

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

267. end;

268. end;

269.

270. procedure TF\_HutangData.ButtonHapusClick(Sender: TObject);

271. begin

272.

273. if ListViewData.Selected = nil then

274. begin

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

276. Exit;

277. end;

278.

279. if not ZConnection.Connected then

280. begin

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

282. Exit;

283. end;

284.

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

286. begin

287. try

288. ZQuery.SQL.Text := 'DELETE FROM hutangdata WHERE id\_hutang = :id';

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

290. ZQuery.ExecSQL;

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

292. RefreshListView;

293. SetStateAwal;

294. except

295. on E: Exception do

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

297. end;

298. end;

299. end;

300.

301. procedure TF\_HutangData.ButtonUbahClick(Sender: TObject);

302. begin

303. if ListViewData.Selected = nil then

304. begin

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

306. Exit;

307. end;

308. FMode := 'edit';

309. SetStateEdit;

310. end;

311.

312. procedure TF\_HutangData.ButtonPreviewClick(Sender: TObject);

313. var

314. LFinanceReport: TF\_BarangkuQuickReport;

315. begin

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

317. try

318. LFinanceReport.GenerateReport(DateTimePicker.Date);

319. finally

320. LFinanceReport.Free;

321. end;

322. end;

323.

324. procedure TF\_HutangData.ListViewDataItemClick(Sender: TObject; Item: TListItem; Selected: Boolean);

325. begin

326.

327. if Selected then

328. begin

329. if Item.SubItems[2] = 'Hutang' then

330. ComboBoxJenis.ItemIndex := 0

331. else if Item.SubItems[2] = 'Piutang' then

332. ComboBoxJenis.ItemIndex := 1

333. else

334. ComboBoxJenis.ItemIndex := -1;

335. EditNama.Text := Item.SubItems[1];

336. EditNominal.Text := Item.SubItems[2];

337. ButtonUbah.Enabled := True;

338. ButtonHapus.Enabled := True;

339. end

340. else

341. begin

342. ButtonUbah.Enabled := False;

343. ButtonHapus.Enabled := False;

344. end;

345. end;

346.

347. procedure TF\_HutangData.ButtonBatalClick(Sender: TObject);

348. begin

349. ClearFormFields;

350. SetStateAwal;

351. end;

352.

353. procedure TF\_HutangData.ButtonKeluarClick(Sender: TObject);

354. begin

355. Application.Terminate;

356. end;

357.

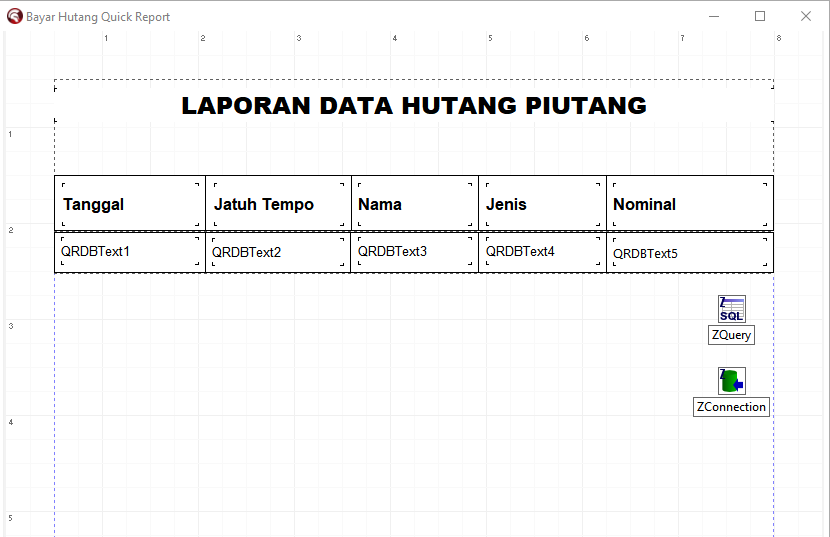
358. end.

359.

## FORM QUICK REPORT

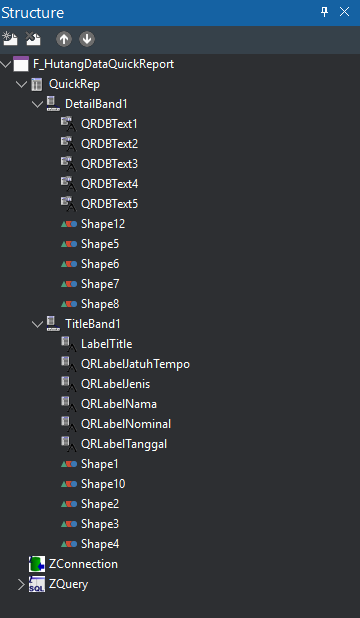
Form Name : F\_HutangDataQuickReport

Form Caption : Bayar Hutang Quick Report



Komponen Form F\_HutangDataQuickReport

|  |  |  |
| --- | --- | --- |
| KOMPONEN | NAME | CAPTION |
| T QuickRep | QuickRep | - |
| TQRBand | DetailBand1 | - |
| TQRBand | TitleBand1 | - |
| TQRDBText | QRDBText1 | - |
| TQRDBText | QRDBText2 | - |
| TQRDBText | QRDBText3 | - |
| TQRDBText | QRDBText4 | - |
| TQRDBText | QRDBText5 | - |
| TQRLabel | LabelTitle | LAPORAN DATA HUTANG PIUTANG |
| TQRLabel | QRLabelJatuhTempo | JatuhTempo |
| TQRLabel | QRLabelJenis | Jenis |
| TQRLabel | QRLabelNama | Nama |
| TQRLabel | QRLabelNominal | Nominal |
| TQRLabel | QRLabelTanggal | Tanggal |
| TQRShape | Shape1 | - |
| TQRShape | Shape2 | - |
| TQRShape | Shape3 | - |
| TQRShape | Shape4 | - |
| TQRShape | Shape5 | - |
| TQRShape | Shape6 | - |
| TQRShape | Shape7 | - |
| TQRShape | Shape8 | - |
| TQRShape | Shape10 | - |
| TQRShape | Shape12 | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |



## SOURCE CODE QUICK REPORT

1. unit HutangDataQuickReport;

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\_HutangDataQuickReport = 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. QRLabelNama: TQRLabel;

23. QRLabelJenis: TQRLabel;

24. QRLabelNominal: 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. QRLabelJatuhTempo: TQRLabel;

36. Shape10: TShape;

37. QRDBText5: TQRDBText;

38. Shape12: TShape;

39. private

40. { Private declarations }

41. procedure ConnectionSetup;

42. procedure QuerySetup(ATanggal: TDateTime);

43. procedure DetailBandSetup;

44. function RupiahFormat(const Value: string): string;

45. procedure RupiahFormatPrint(sender: TObject; var Value: string);

46.

47. public

48. { Public declarations }

49. procedure GenerateReport(ATanggal: TDateTime);

50.

51.

52. end;

53.

54. var

55. F\_HutangDataQuickReport: TF\_HutangDataQuickReport;

56.

57. implementation

58.

59. {$R \*.dfm}

60.

61. procedure TF\_HutangDataQuickReport.ConnectionSetup;

62. begin

63. ZConnection.HostName := 'localhost';

64. ZConnection.Database := 'bayarhutang';

65. ZConnection.User := 'root';

66. ZConnection.Password := '';

67. ZConnection.Protocol := 'mysql';

68. ZConnection.Port := 3306;

69. ZConnection.Connect;

70. ZQuery.Connection := ZConnection;

71. QuickRep.DataSet := ZQuery;

72. end;

73.

74. procedure TF\_HutangDataQuickReport.QuerySetup(ATanggal: TDateTime);

75. begin

76. ZQuery.SQL.Text :=

77. 'SELECT timestamp, jatuh\_tempo, ' +

78. 'CASE WHEN jenis = 0 THEN ''Hutang'' ELSE ''Piutang'' END AS jenis\_str, ' +

79. 'nominal, nama ' +

80. 'FROM hutangdata WHERE DATE(timestamp) = :timestamp';

81. ZQuery.ParamByName('timestamp').AsDate := ATanggal;

82. ZQuery.Open;

83. end;

84.

85. procedure TF\_HutangDataQuickReport.GenerateReport(ATanggal: TDateTime);

86. begin

87. ConnectionSetup;

88. DetailBandSetup;

89. QuerySetup(ATanggal);

90.

91. if ZQuery.IsEmpty then

92. begin

93. ShowMessage('Tidak ada data untuk timestamp yang dipilih.');

94. Exit;

95. end;

96.

97. QuickRep.Preview;

98. end;

99.

100. procedure TF\_HutangDataQuickReport.DetailBandSetup;

101. begin

102. QRDBText1.DataSet := ZQuery;

103. QRDBText2.DataSet := ZQuery;

104. QRDBText3.DataSet := ZQuery;

105. QRDBText4.DataSet := ZQuery;

106. QRDBText5.DataSet := ZQuery;

107. QRDBText1.DataField := 'timestamp';

108. QRDBText2.DataField := 'jatuh\_tempo';

109. QRDBText3.DataField := 'nama';

110. QRDBText4.DataField := 'jenis\_str';

111. QRDBText5.DataField := 'nominal';

112. QRDBText1.Mask := 'd mmmm yyyy';

113. QRDBText2.Mask := 'd mmmm yyyy';

114. QRDBText5.OnPrint := RupiahFormatPrint;

115. end;

116.

117. function TF\_HutangDataQuickReport.RupiahFormat(const Value: string): string;

118. begin

119. // Format as Rupiah, no decimals, no comma, with Rp. prefix

120. try

121. Result := 'Rp. ' + FormatFloat('#,0', StrToFloat(Value));

122. except

123. Result := 'Rp. ' + Value;

124. end;

125. end;

126.

127. procedure TF\_HutangDataQuickReport.RupiahFormatPrint(sender: TObject; var Value: string);

128. begin

129. Value := RupiahFormat(Value);

130. end;

131.

132.

133. end.

134.

# **RINGKASAN DAN KESIMPULAN**

## Ringkasan Proyek

Proyek ini bertujuan untuk mengembangkan sebuah aplikasi manajemen hutang dan piutang, bernama "BayarHutang". Pengembangan aplikasi ini memanfaatkan Delphi sebagai lingkungan pengembangan antarmuka pengguna dan MySQL sebagai sistem manajemen basis data. Aplikasi "BayarHutang" dirancang untuk memfasilitasi pencatatan data hutang dan piutang, meliputi nama, jenis transaksi (hutang atau piutang), nominal, tanggal transaksi, dan tanggal jatuh tempo. Fitur-fitur utama yang diimplementasikan dalam aplikasi ini mencakup operasi dasar manajemen data (Create, Read, Update, Delete - CRUD) serta kapabilitas untuk menghasilkan laporan data hutang dan piutang 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, mendukung pelacakan finansial yang lebih efisien terkait hutang dan piutang.

## 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 := 'bayarhutang';: Menentukan nama database yang akan diakses, yaitu 'bayarhutang'.
  + 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 hutang dan piutang.

## Integrasi dengan QuickReport untuk Pelaporan Data

Untuk menghasilkan laporan hutang dan piutang 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 hutang dan piutang 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 timestamp, jatuh\_tempo, jenis\_str (yang mengonversi nilai numerik jenis menjadi 'Hutang' atau 'Piutang'), nominal, dan nama dari tabel hutangdata, dengan kriteria filter berdasarkan tanggal timestamp. Parameter

:timestamp 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; dan QRDBText1.DataField := 'timestamp';
  + QRDBText2.DataSet := ZQuery; dan QRDBText2.DataField := 'jatuh\_tempo';
  + QRDBText3.DataSet := ZQuery; dan QRDBText3.DataField := 'nama';
  + QRDBText4.DataSet := ZQuery; dan QRDBText4.DataField := 'jenis\_str';
  + QRDBText5.DataSet := ZQuery; dan QRDBText5.DataField := 'nominal';

Pengaturan ini memastikan bahwa setiap

TQRDBText akan menampilkan data yang sesuai dengan pemetaan kolom yang telah ditetapkan.

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

QRDBText1.Mask := 'd mmmm yyyy'; dan QRDBText2.Mask := 'd mmmm yyyy'; memastikan bahwa kolom timestamp dan jatuh\_tempo akan ditampilkan dalam format tanggal yang terstandardisasi (hari bulan tahun). Untuk nominal,

QRDBText5.OnPrint := RupiahFormatPrint; digunakan untuk memformat nilai sebagai mata uang Rupiah dengan awalan "Rp.".

Dengan implementasi langkah-langkah ini, QuickReport dapat secara dinamis mengambil data hutang dan piutang dari TZQuery dan menyajikannya dengan rapi pada laporan, memastikan bahwa setiap komponen TQRDBText menampilkan data yang akurat sesuai dengan pemetaan kolom yang telah didefinisikan.