**PEMROGRAMAN VISUAL**

**IMPLEMENTASI DELPHI & MYSQL  
UNTUK FINANCETRACKER**

TUGAS UAS



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

Disusun Oleh :

Moh Hafiz Anshari 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)

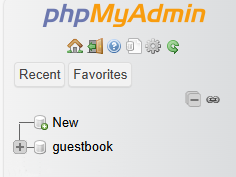
[1. Ringkasan Proyek 17](#_Toc203169653)

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

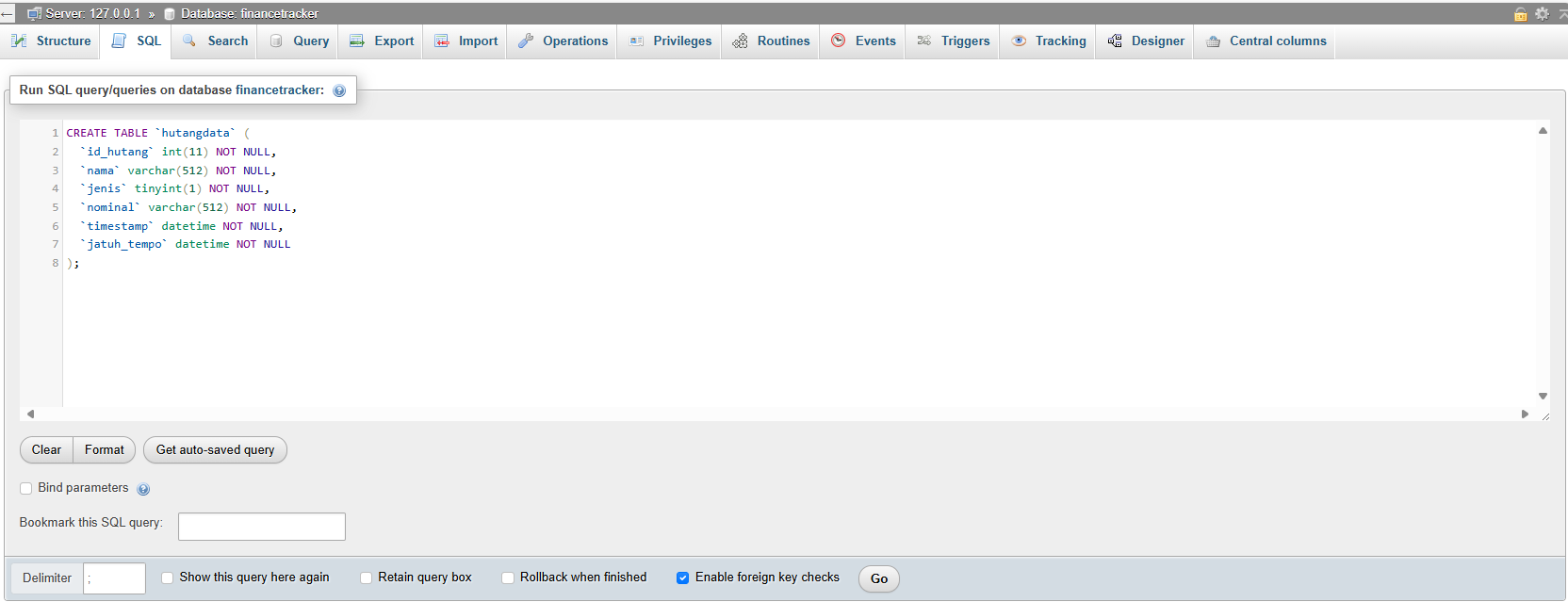
[3. Integrasi dengan QuickReport untuk Pelaporan Data 18](#_Toc203169655)

# **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,

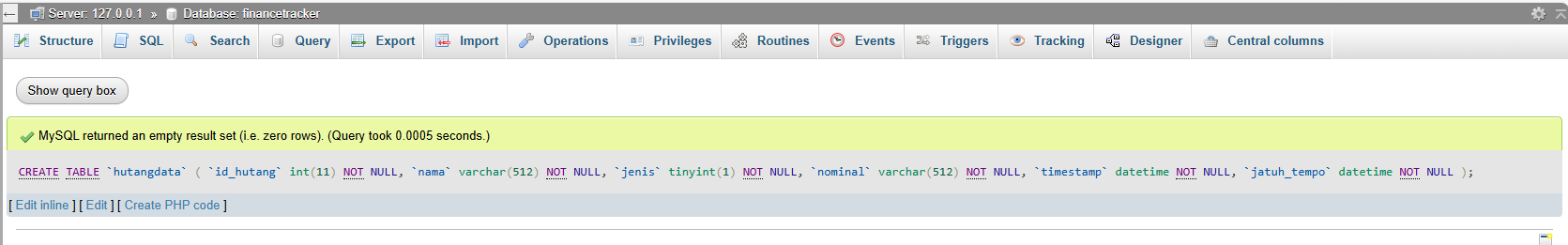
6. `timestamp` datetime NOT NULL,

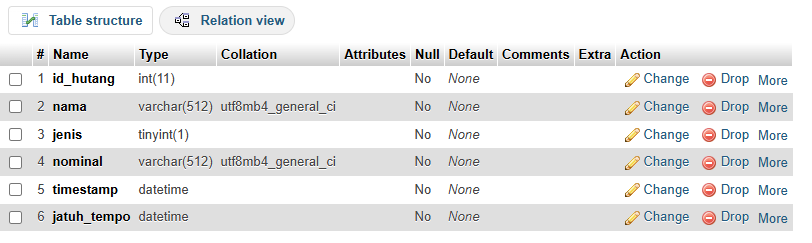
7. `jatuh\_tempo` datetime NOT NULL

8. );

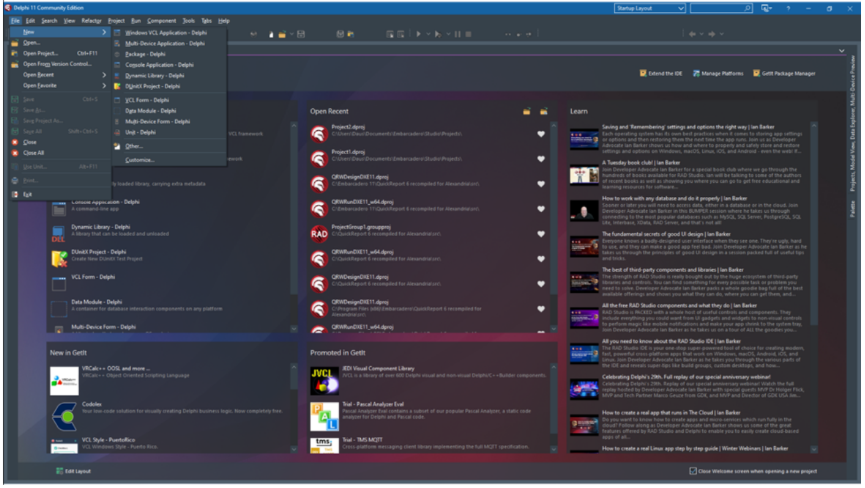
9.

Dan tampilannya akan seperti ini





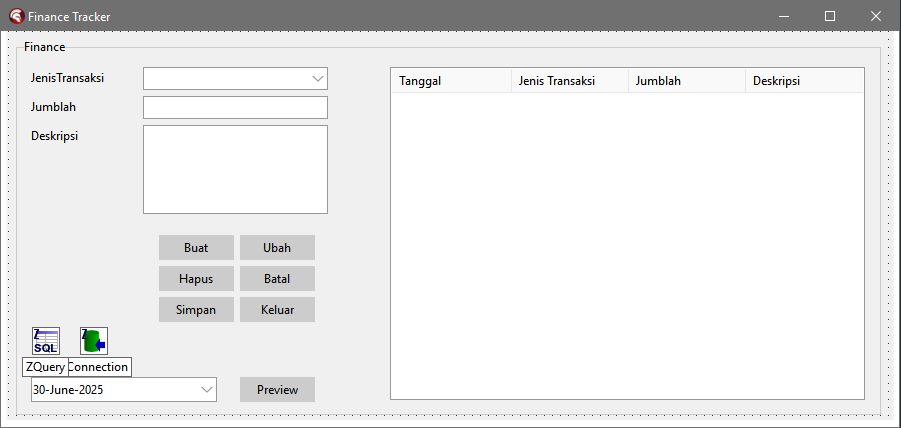
# **PEMBUATAN FORM DELPHI**

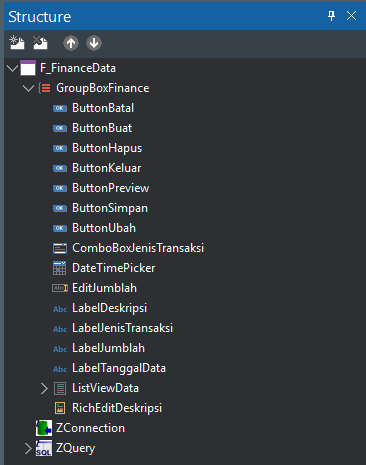
Buka Delphi > File > Windows VCL Application – Delphi

## FORM CRUD DATA

Form Name F\_FinanceData

Form Caption : Finance Tracker





Komponen Form : F\_FinanceData

|  |  |  |
| --- | --- | --- |
| **KOMPONEN** | **NAME** | **CAPTION** |
| TGroupBox | GroupBoxFinance | Finance |
| TButton | ButtonBatal | Batal |
| TButton | ButtonBuat | Buat |
| TButton | ButtonHapus | Hapus |
| TButton | ButtonKeluar | Keluar |
| TButton | ButtonPreview | Preview |
| TButton | ButtonSimpan | Simpan |
| TButton | ButtonUbah | Ubah |
| TComboBox | ComboBoxJenisTransaksi | - |
| TDateTimePicker | DateTimePicker | - |
| TEdit | EditJumblah | - |
| TLabel | LabelDeskripsi | Deskripsi |
| TLabel | LabelJenisTransaksi | JenisTransaksi |
| TLabel | LabelJumblah | Jumblah |
| TLabel | LabelTanggalData | TanggalData |
| TListView | ListViewData | - |
| TRichEdit | RichEditDeskripsi | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |

Struktur ListViewData Column dan aktifkan semua kolom AutoSize

1. Tanggal
2. Jenis Transaksi
3. Jumblah
4. Deskripsi

## SOURCE CODE CRUD DATA

1. unit FinanceData;

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, FinanceDataQuickReport;

10.

11. type

12. TF\_FinanceData = class(TForm)

13. GroupBoxFinance: TGroupBox;

14. ListViewData: TListView;

15. LabelJenisTransaksi: TLabel;

16. LabelDeskripsi: TLabel;

17. RichEditDeskripsi: TRichEdit;

18. ButtonBuat: TButton;

19. ButtonHapus: TButton;

20. ButtonUbah: TButton;

21. ButtonBatal: TButton;

22. ButtonKeluar: TButton;

23. DateTimePicker: TDateTimePicker;

24. LabelTanggalData: TLabel;

25. ButtonPreview: TButton;

26. ZConnection: TZConnection;

27. ZQuery: TZQuery;

28. ButtonSimpan: TButton;

29. ComboBoxJenisTransaksi: TComboBox;

30. EditJumblah: TEdit;

31. LabelJumblah: TLabel;

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\_FinanceData: TF\_FinanceData;

65.

66. implementation

67.

68. {$R \*.dfm}

69.

70. procedure TF\_FinanceData.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\_FinanceData.ClearFormFields;

77. begin

78. ComboBoxJenisTransaksi.ItemIndex := -1;

79. EditJumblah.Clear;

80. RichEditDeskripsi.Lines.Clear;

81. DateTimePicker.Date := Now;

82. ListViewData.Selected := nil;

83. end;

84.

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

86. begin

87. ConfigureMySQLConnection;

88. DateTimePicker.Date := Now;

89. RichEditDeskripsi.Lines.Clear;

90. ComboBoxJenisTransaksi.Items.Clear;

91. ComboBoxJenisTransaksi.Items.Add('Pengeluaran'); // 0

92. ComboBoxJenisTransaksi.Items.Add('Pemasukan'); // 1

93. ComboBoxJenisTransaksi.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\_FinanceData.ConfigureMySQLConnection;

111. begin

112. ZConnection.HostName := 'localhost';

113. ZConnection.Database := 'financetracker';

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\_FinanceData.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. ComboBoxJenisTransaksi.Enabled := False;

134. EditJumblah.Enabled := False;

135. RichEditDeskripsi.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\_FinanceData.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. ComboBoxJenisTransaksi.Enabled := True;

151. EditJumblah.Enabled := True;

152. RichEditDeskripsi.Enabled := True;

153. end;

154.

155. procedure TF\_FinanceData.SetStateInputBaru;

156. begin

157. ButtonBuat.Enabled := False;

158. ButtonSimpan.Enabled := True;

159. ButtonUbah.Enabled := False;

160. ButtonHapus.Enabled := False;

161. ButtonBatal.Enabled := True;

162. ButtonKeluar.Enabled := True;

163. ButtonPreview.Enabled := False;

164. ComboBoxJenisTransaksi.Enabled := True;

165. EditJumblah.Enabled := True;

166. RichEditDeskripsi.Enabled := True;

167. ListViewData.Selected := nil;

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

169. ClearFormFields;

170. end;

171.

172. procedure TF\_FinanceData.RefreshListView;

173. begin

174. ListViewData.Clear;

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

176. ZQuery.Open;

177. while not ZQuery.Eof do

178. begin

179. with ListViewData.Items.Add do

180. begin

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

182. if ZQuery.FieldByName('jenis\_transaksi').AsInteger = 0 then

183. SubItems.Add('Pengeluaran')

184. else

185. SubItems.Add('Pemasukan');

186. SubItems.Add(ZQuery.FieldByName('jumblah').AsString);

187. SubItems.Add(ZQuery.FieldByName('deskripsi').AsString);

188. Data := Pointer(ZQuery.FieldByName('id\_transaksi').AsInteger);

189. end;

190. ZQuery.Next;

191. end;

192. ZQuery.Close;

193. end;

194.

195. procedure TF\_FinanceData.ButtonBuatClick(Sender: TObject);

196. begin

197. FMode := 'create';

198. SetStateInputBaru;

199. end;

200.

201. procedure TF\_FinanceData.ButtonSimpanClick(Sender: TObject);

202. var

203. jenisTransaksi: Integer;

204. begin

205. if (ComboBoxJenisTransaksi.ItemIndex = -1) or

206. (EditJumblah.Text = '') or

207. (RichEditDeskripsi.Lines.Count = 0) then

208. begin

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

210. Exit;

211. end;

212. if not ZConnection.Connected then

213. begin

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

215. Exit;

216. end;

217. jenisTransaksi := ComboBoxJenisTransaksi.ItemIndex; // 0: Pengeluaran, 1: Pemasukan

218. try

219. if FMode = 'create' then

220. begin

221. ZQuery.SQL.Text := 'INSERT INTO financedata (tanggal, jenis\_transaksi, deskripsi, jumblah) VALUES (NOW(), :jenis\_transaksi, :deskripsi, :jumblah)';

222. ZQuery.ParamByName('jenis\_transaksi').AsInteger := jenisTransaksi;

223. ZQuery.ParamByName('deskripsi').AsString := RichEditDeskripsi.Lines.Text;

224. ZQuery.ParamByName('jumblah').AsString := EditJumblah.Text;

225. ZQuery.ExecSQL;

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

227. end

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

229. begin

230. ZQuery.SQL.Text := 'UPDATE financedata SET jenis\_transaksi = :jenis\_transaksi, deskripsi = :deskripsi, jumblah = :jumblah WHERE id\_transaksi = :id\_transaksi';

231. ZQuery.ParamByName('jenis\_transaksi').AsInteger := jenisTransaksi;

232. ZQuery.ParamByName('deskripsi').AsString := RichEditDeskripsi.Lines.Text;

233. ZQuery.ParamByName('jumblah').AsString := EditJumblah.Text;

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

235. ZQuery.ExecSQL;

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

237. end;

238. RefreshListView;

239. SetStateAwal;

240. FMode := '';

241. except

242. on E: Exception do

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

244. end;

245. end;

246.

247. procedure TF\_FinanceData.ButtonHapusClick(Sender: TObject);

248. begin

249.

250. if ListViewData.Selected = nil then

251. begin

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

253. Exit;

254. end;

255.

256. if not ZConnection.Connected then

257. begin

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

259. Exit;

260. end;

261.

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

263. begin

264. try

265. ZQuery.SQL.Text := 'DELETE FROM financedata WHERE id\_transaksi = :id';

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

267. ZQuery.ExecSQL;

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

269. RefreshListView;

270. SetStateAwal;

271. except

272. on E: Exception do

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

274. end;

275. end;

276. end;

277.

278. procedure TF\_FinanceData.ButtonUbahClick(Sender: TObject);

279. begin

280. if ListViewData.Selected = nil then

281. begin

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

283. Exit;

284. end;

285. FMode := 'edit';

286. SetStateEdit;

287. end;

288.

289. procedure TF\_FinanceData.ButtonPreviewClick(Sender: TObject);

290. var

291. LFinanceReport: TF\_FinanceTrackerQuickReport;

292. begin

293. LFinanceReport := TF\_FinanceTrackerQuickReport.Create(Self);

294. try

295. LFinanceReport.GenerateReport(DateTimePicker.Date);

296. finally

297. LFinanceReport.Free;

298. end;

299. end;

300.

301. procedure TF\_FinanceData.ListViewDataItemClick(Sender: TObject; Item: TListItem; Selected: Boolean);

302. begin

303.

304. if Selected then

305. begin

306. if Item.SubItems[0] = 'Pengeluaran' then

307. ComboBoxJenisTransaksi.ItemIndex := 0

308. else if Item.SubItems[0] = 'Pemasukan' then

309. ComboBoxJenisTransaksi.ItemIndex := 1

310. else

311. ComboBoxJenisTransaksi.ItemIndex := -1;

312. EditJumblah.Text := Item.SubItems[1];

313. RichEditDeskripsi.Lines.Text := Item.SubItems[2];

314. ButtonUbah.Enabled := True;

315. ButtonHapus.Enabled := True;

316. end

317. else

318. begin

319. ButtonUbah.Enabled := False;

320. ButtonHapus.Enabled := False;

321. end;

322. end;

323.

324. procedure TF\_FinanceData.ButtonBatalClick(Sender: TObject);

325. begin

326. ClearFormFields;

327. SetStateAwal;

328. end;

329.

330. procedure TF\_FinanceData.ButtonKeluarClick(Sender: TObject);

331. begin

332. Application.Terminate;

333. end;

334.

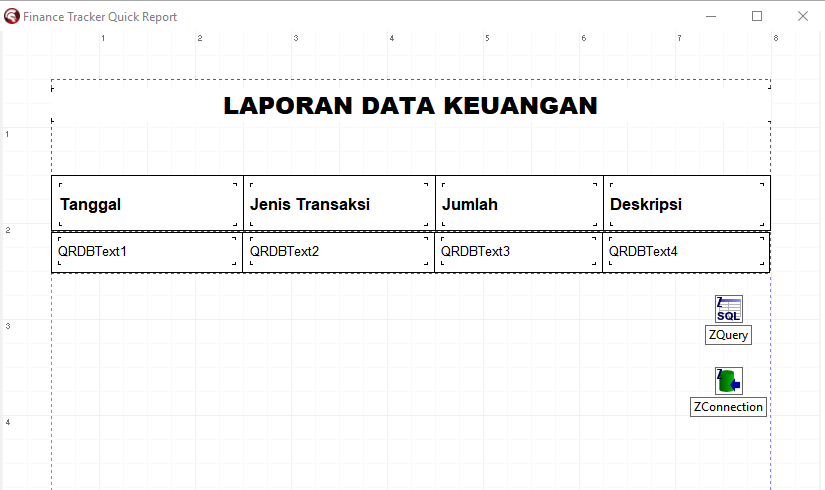
335. end.

336.

## FORM QUICK REPORT

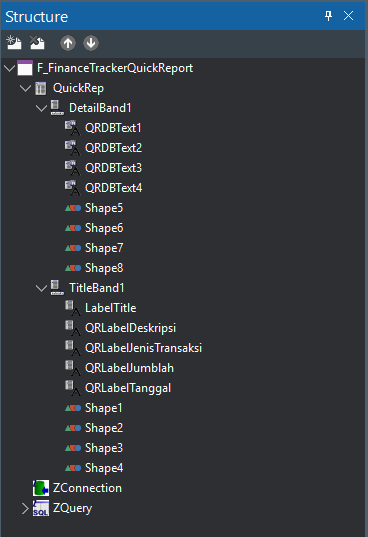
Form Name : F\_FinanceTrackerQuickReport

Form Caption : Finance Tracker Quick Report



Komponen Form F\_FinanceTrackerQuickReport

|  |  |  |
| --- | --- | --- |
| KOMPONEN | NAME | CAPTION |
| T QuickRep | QuickRep | - |
| TQRBand | DetailBand1 | - |
| TQRBand | TitleBand1 | - |
| TQRDBText | QRDBText1 | - |
| TQRDBText | QRDBText2 | - |
| TQRDBText | QRDBText3 | - |
| TQRDBText | QRDBText4 | - |
| TQRLabel | LabelTitle | - |
| TQRLabel | QRLabelDeskripsi | Deskripsi |
| TQRLabel | QRLabelJenis Transaksi | Jenis Transaksi |
| TQRLabel | QRLabelJumblah | Jumblah |
| TQRLabel | QRLabelTanggal | Tanggal |
| TQRShape | Shape1 | - |
| TQRShape | Shape2 | - |
| TQRShape | Shape3 | - |
| TQRShape | Shape4 | - |
| TQRShape | Shape5 | - |
| TQRShape | Shape6 | - |
| TQRShape | Shape7 | - |
| TQRShape | Shape8 | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |



## SOURCE CODE QUICK REPORT

1. unit FinanceDataQuickReport;

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\_FinanceTrackerQuickReport = 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. QRLabelJenisTransaksi: TQRLabel;

23. QRLabelJumblah: TQRLabel;

24. QRLabelDeskripsi: 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. procedure QRDBText3Print(sender: TObject; var Value: string);

41.

42. public

43. { Public declarations }

44. procedure GenerateReport(ATanggal: TDateTime);

45.

46.

47. end;

48.

49. var

50. F\_FinanceTrackerQuickReport: TF\_FinanceTrackerQuickReport;

51.

52. implementation

53.

54. {$R \*.dfm}

55.

56. procedure TF\_FinanceTrackerQuickReport.ConnectionSetup;

57. begin

58. ZConnection.HostName := 'localhost';

59. ZConnection.Database := 'financetracker';

60. ZConnection.User := 'root';

61. ZConnection.Password := '';

62. ZConnection.Protocol := 'mysql';

63. ZConnection.Port := 3306;

64. ZConnection.Connect;

65. ZQuery.Connection := ZConnection;

66. QuickRep.DataSet := ZQuery;

67. end;

68.

69. procedure TF\_FinanceTrackerQuickReport.QuerySetup(ATanggal: TDateTime);

70. begin

71. ZQuery.SQL.Text :=

72. 'SELECT tanggal, ' +

73. 'CASE WHEN jenis\_transaksi = 0 THEN ''Pengeluaran'' ELSE ''Pemasukan'' END AS jenis\_transaksi\_str, ' +

74. 'jumblah, deskripsi ' +

75. 'FROM financedata WHERE DATE(tanggal) = :tanggal';

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

77. ZQuery.Open;

78. end;

79.

80. procedure TF\_FinanceTrackerQuickReport.GenerateReport(ATanggal: TDateTime);

81. begin

82. ConnectionSetup;

83. DetailBandSetup;

84. QuerySetup(ATanggal);

85.

86. if ZQuery.IsEmpty then

87. begin

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

89. Exit;

90. end;

91.

92. QuickRep.Preview;

93. end;

94.

95. procedure TF\_FinanceTrackerQuickReport.DetailBandSetup;

96. begin

97. QRDBText1.DataSet := ZQuery;

98. QRDBText2.DataSet := ZQuery;

99. QRDBText3.DataSet := ZQuery;

100. QRDBText4.DataSet := ZQuery;

101. QRDBText1.DataField := 'tanggal';

102. QRDBText2.DataField := 'jenis\_transaksi\_str';

103. QRDBText3.DataField := 'jumblah';

104. QRDBText4.DataField := 'deskripsi';

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

106. QRDBText3.OnPrint := Self.QRDBText3Print;

107. end;

108.

109. procedure TF\_FinanceTrackerQuickReport.QRDBText3Print(sender: TObject; var Value: string);

110. begin

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

112. try

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

114. except

115. Value := 'Rp. ' + Value;

116. end;

117. end;

118.

119. end.

120.

# **RINGKASAN DAN KESIMPULAN**

## Ringkasan Proyek

Proyek ini bertujuan untuk mengembangkan sebuah aplikasi pengelolaan keuangan pribadi, bernama "FinanceTracker". Pengembangan aplikasi ini memanfaatkan Delphi sebagai lingkungan pengembangan antarmuka pengguna dan MySQL sebagai sistem manajemen basis data. Aplikasi "FinanceTracker" dirancang untuk memfasilitasi pencatatan transaksi keuangan, meliputi jenis transaksi (pengeluaran atau pemasukan), jumlah nominal, dan deskripsi 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 transaksi keuangan 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 keuangan yang lebih efektif.

## 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 := 'financetracker';: Menentukan nama database yang akan diakses, yaitu 'financetracker'.
  + 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 transaksi keuangan.

## Integrasi dengan QuickReport untuk Pelaporan Data

Untuk menghasilkan laporan transaksi keuangan 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 transaksi 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 transaksi yang akan dilaporkan. Dalam prosedur

QuerySetup(ATanggal: TDateTime), properti ZQuery.SQL.Text diatur untuk memilih kolom tanggal, jenis\_transaksi\_str, jumblah, dan deskripsi dari tabel financedata, 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 transaksi), QRDBText3 (jumlah), dan QRDBText4 (deskripsi). Pengaturan ini memastikan bahwa

QRDBText1 akan menampilkan data dari kolom tanggal, QRDBText2 dari jenis\_transaksi\_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). Selain itu,

QRDBText3 memiliki penanganan OnPrint untuk memformat jumblah sebagai mata uang Rupiah.

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

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