**PEMROGRAMAN VISUAL**

**IMPLEMENTASI DELPHI & MYSQL  
UNTUK SMARTCLASS**

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](#_Toc203088633)

[ MEMBUAT DATABASE 1](#_Toc203088634)

[**II.** **PEMBUATAN FORM DELPHI** 3](#_Toc203088635)

[ FORM CRUD DATA 4](#_Toc203088636)

[ SOURCE CODE CRUD DATA 6](#_Toc203088637)

[ FORM QUICK REPORT 14](#_Toc203088638)

[ SOURCE CODE QUICK REPORT 16](#_Toc203088639)

[**III.** **RINGKASAN DAN KESIMPULAN** 18](#_Toc203088640)

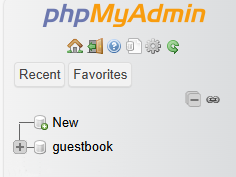
[1. Ringkasan Proyek 18](#_Toc203088641)

[2. Mekanisme Koneksi Database 18](#_Toc203088642)

[3. Integrasi dengan QuickReport untuk Pelaporan Data 19](#_Toc203088643)

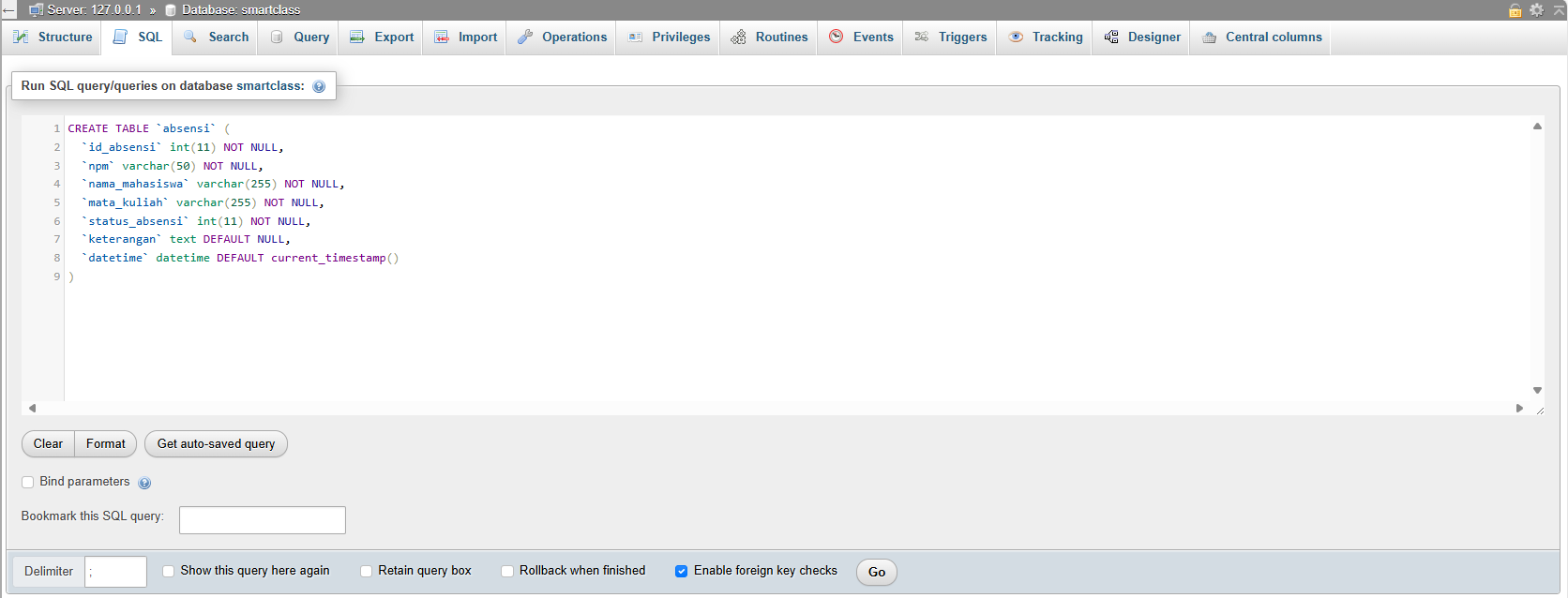
# **PEMBUATAN DATABASE**

## 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 `absensi` (

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

3. `npm` varchar(50) NOT NULL,

4. `nama\_mahasiswa` varchar(255) NOT NULL,

5. `mata\_kuliah` varchar(255) NOT NULL,

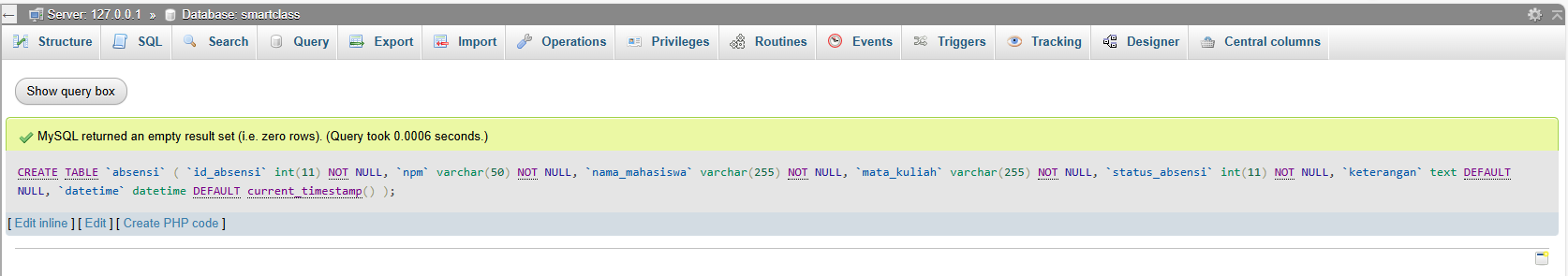
6. `status\_absensi` int(11) NOT NULL,

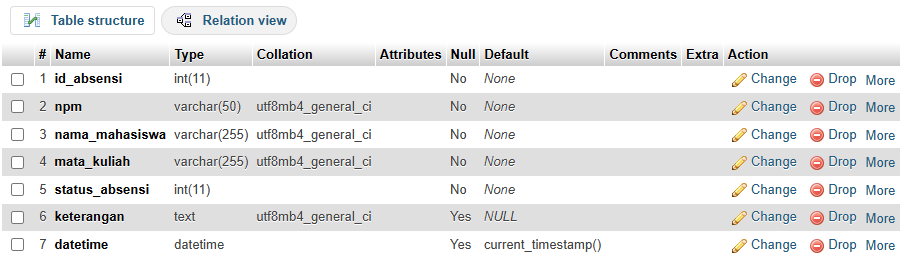
7. `keterangan` text DEFAULT NULL,

8. `datetime` datetime DEFAULT current\_timestamp()

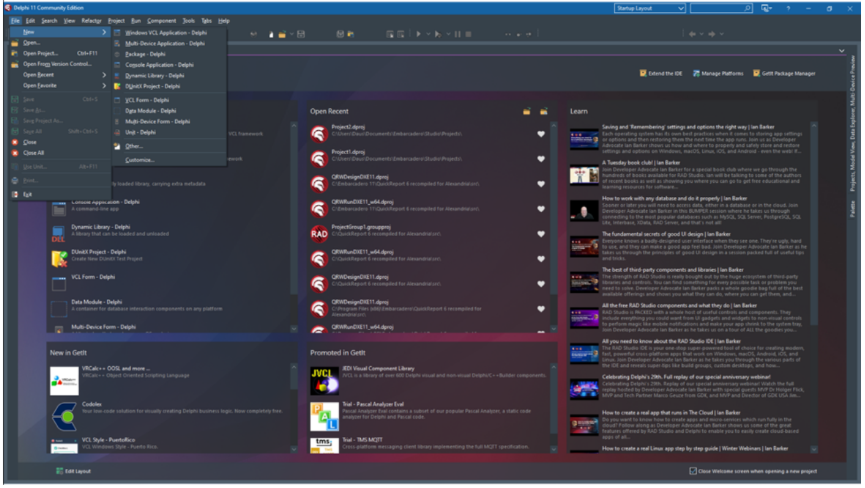
9. )

Dan tampilannya akan seperti ini





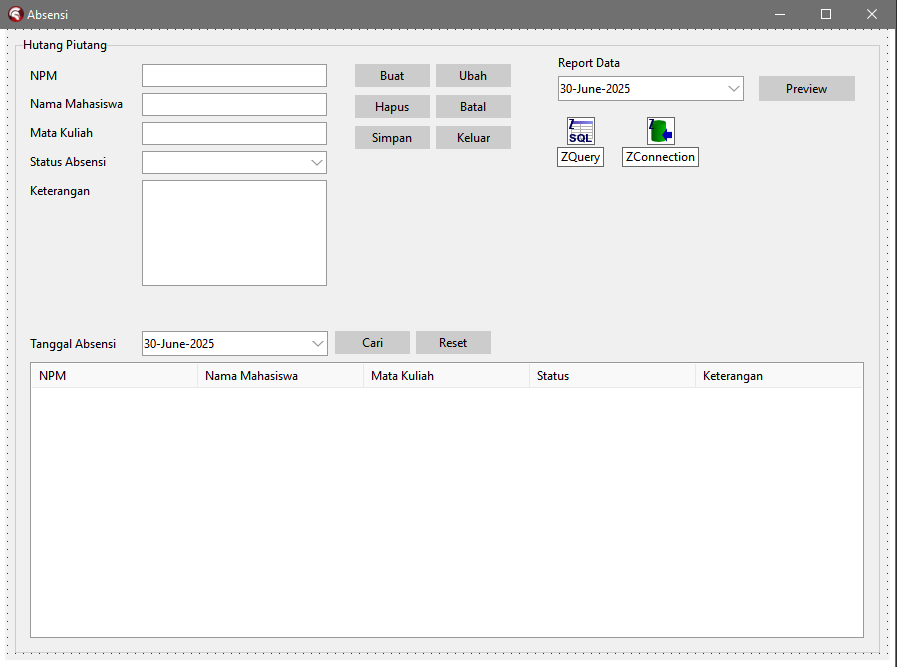
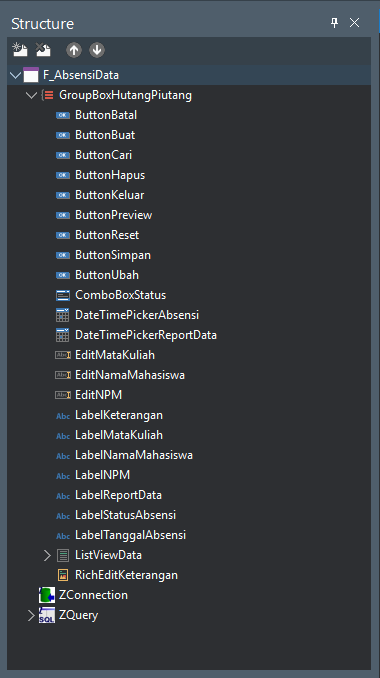
# **PEMBUATAN FORM DELPHI**

Buka Delphi > File > Windows VCL Application – Delphi

## FORM CRUD DATA

Form Name : F\_AbsensiData

Form Caption : Abseni



Komponen Form F\_AbsensiData

|  |  |  |
| --- | --- | --- |
| **KOMPONEN** | **NAME** | **CAPTION** |
| TGroupBox | GroupBoxHutangPiutang | - |
| TButton | ButtonBatal | Batal |
| TButton | ButtonBuat | Buat |
| TButton | ButtonCari | Cari |
| TButton | ButtonHapus | Hapus |
| TButton | ButtonKeluar | Keluar |
| TButton | ButtonPreview | Preview |
| TButton | ButtonReset | Reset |
| TButton | ButtonSimpan | Simpan |
| TButton | ButtonUbah | Ubah |
| TComboBox | ComboBoxStatus | - |
| TDateTimePicker | DateTimePickerAbsensi | - |
| TDateTimePicker | DateTimePickerReportData | - |
| TEdit | EditMataKuliah | - |
| TEdit | EditNamaMahasiswa | - |
| TEdit | EditNPM | - |
| TLabel | LabelKeterangan | Keterangan |
| TLabel | LabelMataKuliah | Mata Kuliah |
| TLabel | LabelNamaMahasiswa | Nama Mahasiswa |
| TLabel | LabelNPM | NPM |
| TLabel | LabelReportData | Report Data |
| TLabel | LabelStatusAbsensi | Status Absensi |
| TLabel | LabelTanggalAbsensi | Tanggal Absensi |
| TListView | ListViewData | - |
| TRichEdit | RichEditKeterangan | - |
| TZConnection | ZConnection | - |
| TZQuery | ZQuery | - |

Struktur ListViewData Column dan aktifkan semua kolom AutoSize

1. NPM
2. Nama Mahasiswa
3. Mata Kuliah
4. Status
5. Keterangan

## SOURCE CODE CRUD DATA

1. unit AbsensiData;

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, AbsensiQuickReport;

10.

11. type

12. TF\_AbsensiData = class(TForm)

13. GroupBoxHutangPiutang: TGroupBox;

14. ListViewData: TListView;

15. LabelNPM: TLabel;

16. LabelNamaMahasiswa: TLabel;

17. ButtonBuat: TButton;

18. ButtonHapus: TButton;

19. ButtonUbah: TButton;

20. ButtonBatal: TButton;

21. ButtonKeluar: TButton;

22. DateTimePickerReportData: TDateTimePicker;

23. LabelReportData: TLabel;

24. ButtonPreview: TButton;

25. ZConnection: TZConnection;

26. ZQuery: TZQuery;

27. ButtonSimpan: TButton;

28. ComboBoxStatus: TComboBox;

29. EditMataKuliah: TEdit;

30. LabelMataKuliah: TLabel;

31. LabelStatusAbsensi: TLabel;

32. LabelKeterangan: TLabel;

33. EditNPM: TEdit;

34. EditNamaMahasiswa: TEdit;

35. RichEditKeterangan: TRichEdit;

36. LabelTanggalAbsensi: TLabel;

37. DateTimePickerAbsensi: TDateTimePicker;

38. ButtonCari: TButton;

39. ButtonReset: TButton;

40.

41. procedure FormCreate(Sender: TObject);

42.

43.

44. procedure ButtonBuatClick(Sender: TObject);

45. procedure ButtonHapusClick(Sender: TObject);

46. procedure ButtonUbahClick(Sender: TObject);

47. procedure ButtonBatalClick(Sender: TObject);

48. procedure ButtonKeluarClick(Sender: TObject);

49. procedure ButtonPreviewClick(Sender: TObject);

50. procedure ButtonSimpanClick(Sender: TObject);

51.

52. procedure ButtonCariClick(Sender: TObject);

53. procedure ButtonResetClick(Sender: TObject);

54.

55.

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

57.

58. // sanitasi agar hanya angka yang dapat di input kepada EditNPM

59. procedure EditNPMKeyPress(Sender: TObject; var Key: Char);

60.

61. procedure ConfigureMySQLConnection;

62. private

63. { Private declarations }

64. FMode: string;

65. procedure SetStateAwal;

66. procedure SetStateEdit;

67. procedure SetStateInputBaru;

68. procedure RefreshListView;

69. procedure ClearFormFields;

70. public

71. { Public declarations }

72. end;

73.

74.

75. var

76. F\_AbsensiData: TF\_AbsensiData;

77.

78. implementation

79.

80. {$R \*.dfm}

81.

82. // Memunculkan data ke ListView berdasarkan tanggal yang dipilih pada DateTimePickerAbsensi

83. procedure TF\_AbsensiData.ButtonCariClick(Sender: TObject);

84. begin

85. if not ZConnection.Connected then

86. begin

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

88. Exit;

89. end;

90.

91. ZQuery.SQL.Text := 'SELECT \* FROM absensi WHERE DATE(datetime) = :tanggal ORDER BY datetime DESC';

92. ZQuery.ParamByName('tanggal').AsDate := DateTimePickerAbsensi.Date;

93. ZQuery.Open;

94.

95. // Check if there are results

96. if ZQuery.IsEmpty then

97. begin

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

99. ZQuery.Close;

100. Exit;

101. end;

102. ListViewData.Clear;

103. while not ZQuery.Eof do

104. begin

105. with ListViewData.Items.Add do

106. begin

107. Caption := ZQuery.FieldByName('npm').AsString;

108. SubItems.Add(ZQuery.FieldByName('nama\_mahasiswa').AsString);

109. SubItems.Add(ZQuery.FieldByName('mata\_kuliah').AsString);

110. if ZQuery.FieldByName('status\_absensi').AsInteger = 0 then

111. SubItems.Add('Hadir')

112. else if ZQuery.FieldByName('status\_absensi').AsInteger = 1 then

113. SubItems.Add('Izin')

114. else if ZQuery.FieldByName('status\_absensi').AsInteger = 2 then

115. SubItems.Add('Sakit')

116. else

117. SubItems.Add('Alpha');

118. SubItems.Add(ZQuery.FieldByName('keterangan').AsString);

119. Data := Pointer(ZQuery.FieldByName('id\_absensi').AsInteger);

120. end;

121. ZQuery.Next;

122. end;

123. ZQuery.Close;

124. ListViewData.Selected := nil; // Clear selection after search

125. SetStateAwal; // Reset state after search

126. end;

127.

128. // Mengembalikan ListView ke kondisi awal dengan data hari ini

129. procedure TF\_AbsensiData.ButtonResetClick(Sender: TObject);

130. begin

131. if not ZConnection.Connected then

132. begin

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

134. Exit;

135. end;

136.

137. DateTimePickerAbsensi.Date := Now; // Reset DateTimePicker to today

138. ZQuery.SQL.Text := 'SELECT \* FROM absensi WHERE DATE(datetime) = :tanggal ORDER BY datetime DESC';

139. ZQuery.ParamByName('tanggal').AsDate := DateTimePickerAbsensi.Date;

140. ZQuery.Open;

141.

142. ListViewData.Clear;

143. while not ZQuery.Eof do

144. begin

145. with ListViewData.Items.Add do

146. begin

147. Caption := ZQuery.FieldByName('npm').AsString;

148. SubItems.Add(ZQuery.FieldByName('nama\_mahasiswa').AsString);

149. SubItems.Add(ZQuery.FieldByName('mata\_kuliah').AsString);

150. if ZQuery.FieldByName('status\_absensi').AsInteger = 0 then

151. SubItems.Add('Hadir')

152. else if ZQuery.FieldByName('status\_absensi').AsInteger = 1 then

153. SubItems.Add('Izin')

154. else if ZQuery.FieldByName('status\_absensi').AsInteger = 2 then

155. SubItems.Add('Sakit')

156. else

157. SubItems.Add('Alpha');

158. SubItems.Add(ZQuery.FieldByName('keterangan').AsString);

159. Data := Pointer(ZQuery.FieldByName('id\_absensi').AsInteger);

160. end;

161. ZQuery.Next;

162. end;

163. ZQuery.Close;

164.

165. ClearFormFields; // Clear form fields after reset

166. end;

167.

168.

169.

170. procedure TF\_AbsensiData.EditNPMKeyPress(Sender: TObject; var Key: Char);

171. begin

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

173. Key := #0;

174. end;

175.

176. procedure TF\_AbsensiData.ClearFormFields;

177. begin

178. ComboBoxStatus.ItemIndex := -1;

179. EditNPM.Clear;

180. EditNamaMahasiswa.Clear;

181. EditMataKuliah.Clear;

182. RichEditKeterangan.Clear;

183. DateTimePickerReportData.Date := Now;

184. DateTimePickerAbsensi.Date := Now;

185. ListViewData.Selected := nil;

186. end;

187.

188. procedure TF\_AbsensiData.FormCreate(Sender: TObject);

189. begin

190. ConfigureMySQLConnection;

191. ComboBoxStatus.Items.Clear;

192. ComboBoxStatus.Items.Add('Hadir'); // 0

193. ComboBoxStatus.Items.Add('Izin'); // 1

194. ComboBoxStatus.Items.Add('Sakit'); // 2

195. ComboBoxStatus.Items.Add('Alpha'); // 3

196. ClearFormFields;

197. RefreshListView;

198. ListViewData.OnSelectItem := ListViewDataItemClick;

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

200. ListViewData.ViewStyle := vsReport;

201. ListViewData.RowSelect := True;

202. ButtonBuat.OnClick := ButtonBuatClick;

203. ButtonHapus.OnClick := ButtonHapusClick;

204. ButtonUbah.OnClick := ButtonUbahClick;

205. ButtonBatal.OnClick := ButtonBatalClick;

206. ButtonKeluar.OnClick := ButtonKeluarClick;

207. ButtonPreview.OnClick := ButtonPreviewClick;

208. ButtonSimpan.OnClick := ButtonSimpanClick;

209. EditNPM.OnKeyPress := EditNPMKeyPress;

210. ButtonCari.OnClick := ButtonCariClick;

211. ButtonReset.OnClick := ButtonResetClick;

212. SetStateAwal;

213. end;

214.

215. procedure TF\_AbsensiData.ConfigureMySQLConnection;

216. begin

217. ZConnection.HostName := 'localhost';

218. ZConnection.Database := 'smartclass';

219. ZConnection.User := 'root';

220. ZConnection.Password := '';

221. ZConnection.Protocol := 'mysql';

222. ZConnection.Port := 3306;

223. ZConnection.Connect;

224. ZQuery.Connection := ZConnection;

225. if not ZConnection.Connected then

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

227. end;

228.

229. procedure TF\_AbsensiData.SetStateAwal;

230. begin

231. ButtonBuat.Enabled := True;

232. ButtonSimpan.Enabled := False;

233. ButtonUbah.Enabled := False;

234. ButtonHapus.Enabled := False;

235. ButtonBatal.Enabled := True;

236. ButtonKeluar.Enabled := True;

237. ButtonPreview.Enabled := True;

238.

239. EditNPM.Enabled := False;

240. EditMataKuliah.Enabled := False;

241. EditNamaMahasiswa.Enabled := False;

242. ComboBoxStatus.Enabled := False;

243. RichEditKeterangan.Enabled := False;

244.

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

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

247. ClearFormFields;

248. end;

249.

250. procedure TF\_AbsensiData.SetStateEdit;

251. begin

252. ButtonBuat.Enabled := False;

253. ButtonSimpan.Enabled := True;

254. ButtonUbah.Enabled := True;

255. ButtonHapus.Enabled := True;

256. ButtonBatal.Enabled := True;

257. ButtonKeluar.Enabled := True;

258. ButtonPreview.Enabled := True;

259.

260. EditNPM.Enabled := True;

261. EditMataKuliah.Enabled := True;

262. EditNamaMahasiswa.Enabled := True;

263. ComboBoxStatus.Enabled := True;

264. RichEditKeterangan.Enabled := True;

265.

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

267. end;

268.

269. procedure TF\_AbsensiData.SetStateInputBaru;

270. begin

271. ButtonBuat.Enabled := False;

272. ButtonSimpan.Enabled := True;

273. ButtonUbah.Enabled := False;

274. ButtonHapus.Enabled := False;

275. ButtonBatal.Enabled := True;

276. ButtonKeluar.Enabled := True;

277. ButtonPreview.Enabled := False;

278.

279. EditNPM.Enabled := True;

280. EditMataKuliah.Enabled := True;

281. EditNamaMahasiswa.Enabled := True;

282. ComboBoxStatus.Enabled := True;

283. RichEditKeterangan.Enabled := True;

284.

285. ListViewData.Selected := nil;

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

287. ClearFormFields;

288. end;

289.

290. procedure TF\_AbsensiData.RefreshListView;

291. begin

292. ListViewData.Clear;

293. ZQuery.SQL.Text := 'SELECT \* FROM absensi WHERE DATE(datetime) = :tanggal ORDER BY datetime DESC';

294. ZQuery.ParamByName('tanggal').AsDate := Date;

295. ZQuery.Open;

296. while not ZQuery.Eof do

297. begin

298. with ListViewData.Items.Add do

299. begin

300. Caption := ZQuery.FieldByName('npm').AsString;

301. SubItems.Add(ZQuery.FieldByName('nama\_mahasiswa').AsString);

302. SubItems.Add(ZQuery.FieldByName('mata\_kuliah').AsString);

303. if ZQuery.FieldByName('status\_absensi').AsInteger = 0 then

304. SubItems.Add('Hadir')

305. else if ZQuery.FieldByName('status\_absensi').AsInteger = 1 then

306. SubItems.Add('Izin')

307. else if ZQuery.FieldByName('status\_absensi').AsInteger = 2 then

308. SubItems.Add('Sakit')

309. else

310. SubItems.Add('Alpha');

311. SubItems.Add(ZQuery.FieldByName('keterangan').AsString);

312. Data := Pointer(ZQuery.FieldByName('id\_absensi').AsInteger);

313. end;

314. ZQuery.Next;

315. end;

316. ZQuery.Close;

317. end;

318.

319. procedure TF\_AbsensiData.ButtonBuatClick(Sender: TObject);

320. begin

321. FMode := 'create';

322. SetStateInputBaru;

323. end;

324.

325. procedure TF\_AbsensiData.ButtonSimpanClick(Sender: TObject);

326. var

327. indexComboBox: Integer;

328. begin

329. if (ComboBoxStatus.ItemIndex = -1) or

330. (EditNPM.Text = '') or

331. (EditMataKuliah.Text = '') or

332. (EditNamaMahasiswa.Text = '') then

333. begin

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

335. Exit;

336. end

337. else if (ComboBoxStatus.ItemIndex > 0) and (RichEditKeterangan.Text = '') then

338. begin

339. ShowMessage('Silakan isi keterangan jika status absensi tidak hadir.');

340. Exit;

341. end;

342. if not ZConnection.Connected then

343. begin

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

345. Exit;

346. end;

347. indexComboBox := ComboBoxStatus.ItemIndex; // 0: Hadir, 1: Izin, 2: Sakit, 3: Alpha

348. try

349. if FMode = 'create' then

350. begin

351. ZQuery.SQL.Text := 'INSERT INTO absensi (datetime, npm, nama\_mahasiswa, mata\_kuliah, status\_absensi, keterangan) ' +

352. 'VALUES (:datetime, :npm, :nama\_mahasiswa, :mata\_kuliah, :status\_absensi, :keterangan)';

353. ZQuery.ParamByName('datetime').AsDateTime := Now;

354. ZQuery.ParamByName('npm').AsString := EditNPM.Text;

355. ZQuery.ParamByName('nama\_mahasiswa').AsString := EditNamaMahasiswa.Text;

356. ZQuery.ParamByName('mata\_kuliah').AsString := EditMataKuliah.Text;

357. ZQuery.ParamByName('status\_absensi').AsInteger := indexComboBox;

358. if (indexComboBox = 0) and (RichEditKeterangan.Text = '') then

359. ZQuery.ParamByName('keterangan').AsString := '-'

360. else

361. ZQuery.ParamByName('keterangan').AsString := RichEditKeterangan.Text;

362. ZQuery.ExecSQL;

363. ShowMessage('Data absensi berhasil ditambahkan.');

364. end

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

366. begin

367. ZQuery.SQL.Text := 'UPDATE absensi SET ' +

368. 'status\_absensi = :status\_absensi, ' +

369. 'nama\_mahasiswa = :nama\_mahasiswa, ' +

370. 'keterangan = :keterangan ' +

371. 'WHERE id\_absensi = :id\_absensi';

372. ZQuery.ParamByName('status\_absensi').AsInteger := indexComboBox;

373. ZQuery.ParamByName('nama\_mahasiswa').AsString := EditNamaMahasiswa.Text;

374. if (indexComboBox = 0) and (RichEditKeterangan.Text = '') then

375. ZQuery.ParamByName('keterangan').AsString := '-'

376. else

377. ZQuery.ParamByName('keterangan').AsString := RichEditKeterangan.Text;

378. ZQuery.ExecSQL;

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

380. ZQuery.ExecSQL;

381. ShowMessage('Data absensi berhasil diubah.');

382. end;

383. RefreshListView;

384. SetStateAwal;

385. FMode := '';

386. except

387. on E: Exception do

388. ShowMessage('Gagal menyimpan data absensi: ' + E.Message);

389. end;

390. end;

391.

392. procedure TF\_AbsensiData.ButtonHapusClick(Sender: TObject);

393. begin

394.

395. if ListViewData.Selected = nil then

396. begin

397. ShowMessage('Silakan pilih data absensi yang ingin dihapus.');

398. Exit;

399. end;

400.

401. if not ZConnection.Connected then

402. begin

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

404. Exit;

405. end;

406.

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

408. begin

409. try

410. ZQuery.SQL.Text := 'DELETE FROM absensi WHERE id\_absensi = :id';

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

412. ZQuery.ExecSQL;

413. ShowMessage('Data absensi berhasil dihapus.');

414. RefreshListView;

415. SetStateAwal;

416. except

417. on E: Exception do

418. ShowMessage('Gagal menghapus data absensi: ' + E.Message);

419. end;

420. end;

421. end;

422.

423. procedure TF\_AbsensiData.ButtonUbahClick(Sender: TObject);

424. begin

425. if ListViewData.Selected = nil then

426. begin

427. ShowMessage('Silakan pilih data absensi yang ingin diubah.');

428. Exit;

429. end;

430. FMode := 'edit';

431. SetStateEdit;

432. end;

433.

434. procedure TF\_AbsensiData.ButtonPreviewClick(Sender: TObject);

435. var

436. LFinanceReport: TF\_AbsensiQuickReport;

437. begin

438. LFinanceReport := TF\_AbsensiQuickReport.Create(Self);

439. try

440. LFinanceReport.GenerateReport(DateTimePickerReportData.Date);

441. finally

442. LFinanceReport.Free;

443. end;

444. end;

445.

446. procedure TF\_AbsensiData.ListViewDataItemClick(Sender: TObject; Item: TListItem; Selected: Boolean);

447. begin

448.

449. if Selected then

450. begin

451. if Item.SubItems[2] = 'Hadir' then

452. ComboBoxStatus.ItemIndex := 0

453. else if Item.SubItems[2] = 'Izin' then

454. ComboBoxStatus.ItemIndex := 1

455. else if Item.SubItems[2] = 'Sakit' then

456. ComboBoxStatus.ItemIndex := 2

457. else if Item.SubItems[2] = 'Alpha' then

458. ComboBoxStatus.ItemIndex := 3

459. else

460. ComboBoxStatus.ItemIndex := -1;

461. EditNamaMahasiswa.Text := Item.SubItems[0];

462. EditMataKuliah.Text := Item.SubItems[1];

463. RichEditKeterangan.Text := Item.SubItems[3];

464. EditNPM.Text := Item.Caption;

465. ButtonUbah.Enabled := True;

466. ButtonHapus.Enabled := True;

467. end

468. else

469. begin

470. ButtonUbah.Enabled := False;

471. ButtonHapus.Enabled := False;

472. end;

473. end;

474.

475. procedure TF\_AbsensiData.ButtonBatalClick(Sender: TObject);

476. begin

477. ClearFormFields;

478. SetStateAwal;

479. end;

480.

481. procedure TF\_AbsensiData.ButtonKeluarClick(Sender: TObject);

482. begin

483. Application.Terminate;

484. end;

485.

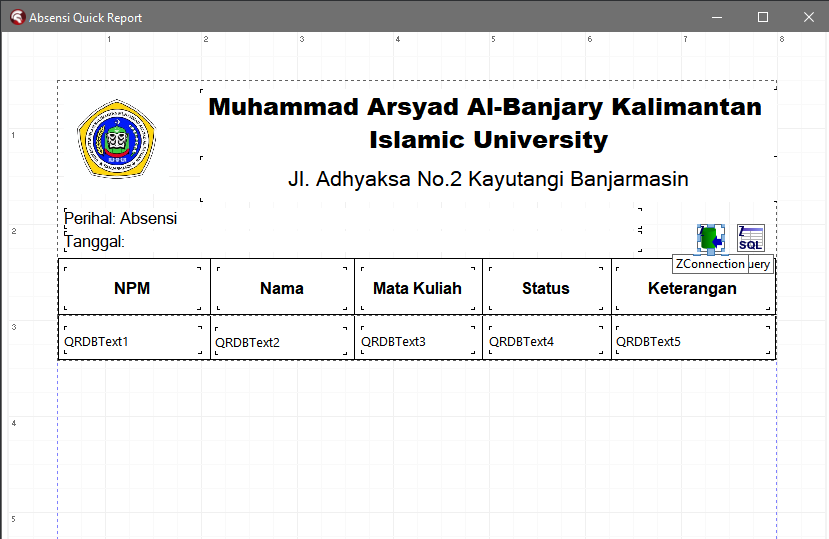
486. end.

487.

## FORM QUICK REPORT

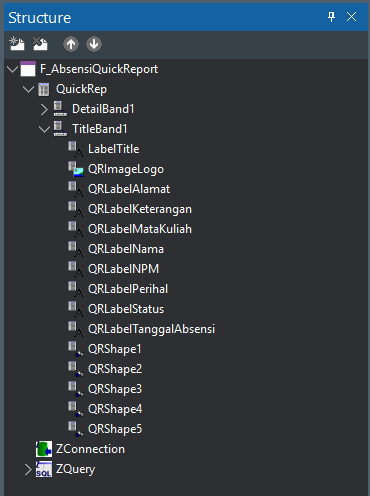
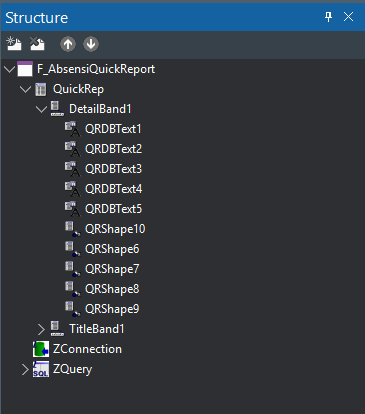
Form Name : F\_AbsensiQuickReport

Form Caption : Absensi Quick Report



Komponen Form F\_AbsensiQuickReport

|  |  |  |
| --- | --- | --- |
| KOMPONEN | NAME | CAPTION |
| T QuickRep | QuickRep | - |
| TQRBand | DetailBand1 | - |
| TQRBand | TitleBand1 | - |
| TQRDBText | QRDBText1 | - |
| TQRDBText | QRDBText2 | - |
| TQRDBText | QRDBText3 | - |
| TQRDBText | QRDBText4 | - |
| TQRDBText | QRDBText5 | - |
| TQRLabel | LabelTitle | Muhammad Arsyad Al-Banjary Kalimantan Islamic University |
| QRImageLogo | QRImageLogo | - |
| TQRLabel | QRLabelAlamat | Jl. Adhyaksa No.2 Kayutangi Banjarmasin |
| TQRLabel | QRLabelKeterangan | Keterangan |
| TQRLabel | QRLabelMataKuliah | Mata Kuliah |
| TQRLabel | QRLabelNama | Nama |
| TQRLabel | QRLabelNPM | NPM |
| TQRLabel | QRLabelPerihal | Perihal: Absensi |
| TQRLabel | QRLabelStatus | Status |
| TQRLabel | QRLabelTanggalAbsensi | Tanggal: |
| TQRShape | QRShape1 | - |
| TQRShape | QRShape2 | - |
| TQRShape | QRShape3 | - |
| TQRShape | QRShape4 | - |
| TQRShape | QRShape5 |  |
| TQRShape | QRShape6 |  |
| TQRShape | QRShape7 |  |
| TQRShape | QRShape8 |  |
| TQRShape | QRShape9 |  |
| TQRShape | QRShape10 |  |
| TZConnection | ZConnection |  |
| TZQuery | ZQuery |  |

## SOURCE CODE QUICK REPORT

1. unit AbsensiQuickReport;

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, Vcl.Imaging.pngimage;

10.

11. type

12. TF\_AbsensiQuickReport = class(TForm)

13. QuickRep: TQuickRep;

14. TitleBand1: TQRBand;

15. DetailBand1: TQRBand;

16. LabelTitle: TQRLabel;

17. QRLabelNPM: TQRLabel;

18. QRLabelMataKuliah: TQRLabel;

19. QRLabelStatus: TQRLabel;

20. QRLabelKeterangan: TQRLabel;

21. QRDBText1: TQRDBText;

22. QRDBText2: TQRDBText;

23. QRDBText3: TQRDBText;

24. QRDBText4: TQRDBText;

25. ZQuery: TZQuery;

26. ZConnection: TZConnection;

27. QRLabelNama: TQRLabel;

28. QRDBText5: TQRDBText;

29. QRImageLogo: TQRImage;

30. QRLabelAlamat: TQRLabel;

31. QRLabelPerihal: TQRLabel;

32. QRLabelTanggalAbsensi: TQRLabel;

33. QRShape1: TQRShape;

34. QRShape2: TQRShape;

35. QRShape3: TQRShape;

36. QRShape4: TQRShape;

37. QRShape5: TQRShape;

38. QRShape6: TQRShape;

39. QRShape7: TQRShape;

40. QRShape8: TQRShape;

41. QRShape9: TQRShape;

42. QRShape10: TQRShape;

43. private

44. { Private declarations }

45. procedure ConnectionSetup;

46. procedure QuerySetup(ATanggal: TDateTime);

47. procedure DetailBandSetup;

48.

49. public

50. { Public declarations }

51. procedure GenerateReport(ATanggal: TDateTime);

52.

53. end;

54.

55. var

56. F\_AbsensiQuickReport: TF\_AbsensiQuickReport;

57.

58. implementation

59.

60. {$R \*.dfm}

61.

62. procedure TF\_AbsensiQuickReport.ConnectionSetup;

63. begin

64. ZConnection.HostName := 'localhost';

65. ZConnection.Database := 'smartclass';

66. ZConnection.User := 'root';

67. ZConnection.Password := '';

68. ZConnection.Protocol := 'mysql';

69. ZConnection.Port := 3306;

70. ZConnection.Connect;

71. ZQuery.Connection := ZConnection;

72. QuickRep.DataSet := ZQuery;

73. end;

74.

75. procedure TF\_AbsensiQuickReport.QuerySetup(ATanggal: TDateTime);

76. begin

77. ZQuery.SQL.Text :=

78. 'SELECT datetime, ' +

79. 'CASE ' +

80. ' WHEN status\_absensi = 0 THEN ''Hadir'' ' +

81. ' WHEN status\_absensi = 1 THEN ''Izin'' ' +

82. ' WHEN status\_absensi = 2 THEN ''Sakit'' ' +

83. ' ELSE ''Alpha'' ' +

84. 'END AS jenis\_str, ' +

85. 'npm, nama\_mahasiswa, mata\_kuliah, keterangan ' +

86. 'FROM absensi WHERE DATE(datetime) = :datetime';

87. ZQuery.ParamByName('datetime').AsDate := ATanggal;

88. ZQuery.Open;

89. end;

90.

91. procedure TF\_AbsensiQuickReport.GenerateReport(ATanggal: TDateTime);

92. begin

93. ConnectionSetup;

94. DetailBandSetup;

95. QuerySetup(ATanggal);

96. QRLabelTanggalAbsensi.Caption := 'Tanggal: ' + FormatDateTime('dd mmmm yyyy', ATanggal);

97.

98. if ZQuery.IsEmpty then

99. begin

100. ShowMessage('Tidak ada data untuk datetime yang dipilih.');

101. Exit;

102. end;

103.

104. QuickRep.Preview;

105. end;

106.

107. procedure TF\_AbsensiQuickReport.DetailBandSetup;

108. begin

109. QRDBText1.DataSet := ZQuery;

110. QRDBText2.DataSet := ZQuery;

111. QRDBText3.DataSet := ZQuery;

112. QRDBText4.DataSet := ZQuery;

113. QRDBText5.DataSet := ZQuery;

114. QRDBText1.DataField := 'npm';

115. QRDBText2.DataField := 'nama\_mahasiswa';

116. QRDBText3.DataField := 'mata\_kuliah';

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

118. QRDBText5.DataField := 'keterangan';

119. end;

120.

121. end.

122.

# **RINGKASAN DAN KESIMPULAN**

## **Ringkasan Proyek**

Proyek ini bertujuan untuk mengembangkan sebuah aplikasi sistem informasi manajemen kelas yang cerdas, bernama "SmartClass". Pengembangan aplikasi ini memanfaatkan Delphi sebagai lingkungan pengembangan antarmuka pengguna dan MySQL sebagai sistem manajemen basis data. Aplikasi "SmartClass" dirancang untuk memfasilitasi pencatatan data absensi mahasiswa, meliputi Nomor Pokok Mahasiswa (NPM), nama mahasiswa, mata kuliah, dan waktu kehadiran. Fitur-fitur utama yang diimplementasikan dalam aplikasi ini mencakup operasi dasar manajemen data (Create, Read, Update, Delete - CRUD) serta kapabilitas untuk menghasilkan laporan data absensi 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 proses administrasi kelas yang lebih efisien.

## 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 := 'smartclass';: Menentukan nama database yang akan diakses, yaitu 'smartclass'.
  + 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 absensi mahasiswa.

## Integrasi dengan QuickReport untuk Pelaporan Data

Untuk menghasilkan laporan absensi 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 absensi 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 absensi yang akan dilaporkan. Dalam prosedur QuerySetup(ATanggal: TDateTime), properti ZQuery.SQL.Text diatur untuk memilih kolom waktu\_kedatangan, npm, nama, dan mata\_kuliah dari tabel absensi, dengan kriteria filter berdasarkan tanggal (WHERE DATE(waktu\_kedatangan) = :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 := 'waktu\_kedatangan'; Konfigurasi serupa diterapkan untuk QRDBText2 (npm), QRDBText3 (nama), dan QRDBText4 (mata kuliah). Pengaturan ini memastikan bahwa QRDBText1 akan menampilkan data dari kolom waktu\_kedatangan, QRDBText2 dari npm, 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 := 'dd/mm/yyyy hh:nn:ss'; memastikan bahwa kolom waktu\_kedatangan 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 absensi dari TZQuery dan menyajikannya dengan rapi pada laporan, memastikan bahwa setiap komponen TQRDBText menampilkan data yang akurat sesuai dengan pemetaan kolom yang telah didefinisikan.