

BAB IV

IMPLEMENTASI

Bab ini berisi tentang implementasi dari Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS berdasarkan analisis kebutuhan dan perancangan sistem maupun perancangan perangkat lunak. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan – batasan dalam implementasi, implementasi basis data, implemmentasi *method*, dan implementasi antarmuka.

4.1 Lingkungan Pengujian Sistem

Perangkat lunak Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Pengujian Perangkat Keras

Tabel 4.1 Lingkungan Pengujian Perangkat Keras

| | |
|-------------------|---|
| <i>Processor</i> | Intel® Core™ i5-450M Processor (2.40 GHz, Cache 3 MB) |
| <i>Memory</i> | 2 GB DDR3 SODIMM PC-8500, Max. Memory 8 GB (2 DIMMs) |
| <i>Hard Drive</i> | 500 GB Serial ATA 5400 RPM |
| <i>Networking</i> | Speed 10 / 100 Mbps |
| <i>Modem</i> | Wavecom Fastrack M1306B USB |

Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS telah diimplementasikan pada sebuah ‘Laptop Acer Aspire 4741’ dengan spesifikasi perangkat keras tertera pada **Tabel 4.1 Lingkungan Pengujian Perangkat Keras**. *Processor* menggunakan Intel® Core™ i5-450M dengan kecepatan 2.40 GHz, Cache 3 MB, *memory* menggunakan DDR SODIMM PC-8500 dengan kapasitas sebesar 2 GB yang dapat diperbesar sampai 8 GB, menggunakan *Hard Drive* Serial ATA 5400 RPM dengan kapasitas sebesar 500

GB, kecepatan *Networking* sebesar 10 / 100 Mbps, dan menggunakan modem Wavecom dengan tipe Fastrack M1306B USB sebagai modem untuk penerima maupun mengirimkan SMS.

4.1.2 Lingkungan Pengujian Perangkat Lunak

Tabel 4.2 Lingkungan Pengujian Perangkat Lunak

| | |
|-----------------------------------|---------------------------|
| <i>Programming Language</i> | <i>C#</i> |
| <i>Database Management System</i> | <i>SQL Server 2008 R2</i> |
| <i>SMS Gateway</i> | <i>AT Commands</i> |

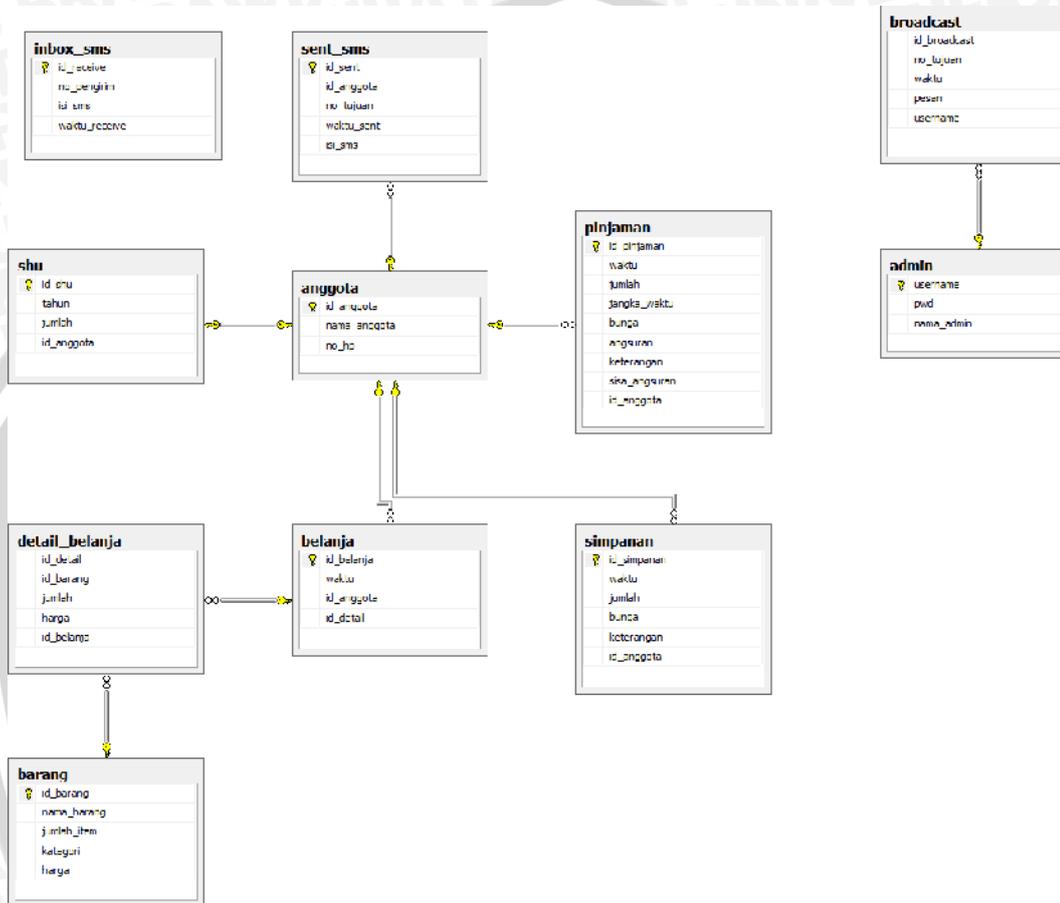
Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS telah diimplementasikan dengan menggunakan perangkat lunak dengan spesifikasi pada **Tabel 4.2 Lingkungan Pengujian Perangkat Lunak**. Menggunakan bahasa pemrograman C#, dengan *database* SQL Server 2008 R2, dan AT Commands sebagai perintah yang diberikan untuk mengirimkan maupun membaca SMS.

4.2 Batasan Implementasi

1. Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS dirancang menggunakan Microsoft Visual Studio 2010 dengan menggunakan bahasa pemrograman C# yang berbasis desktop.
2. Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS menggunakan basis data SQL Server 2008.
3. Perintah SMS menggunakan AT Commands.
4. *Import* data dari excel harus sinkron dengan *database* sql server.

4.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan SQL Server R2. Berdasarkan perancangan basis data **Gambar 3.10 Desain Database** yaitu pada **Gambar 4.1 Implementasi Database**.

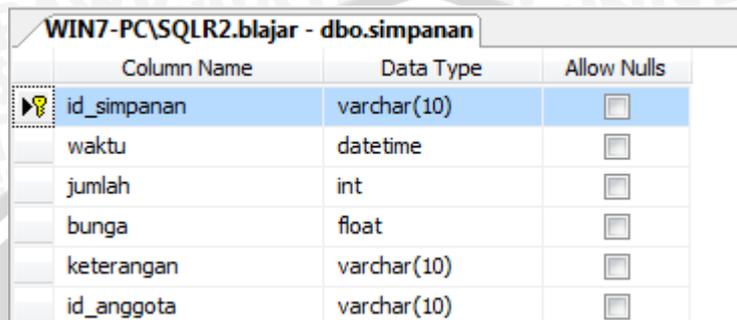


Gambar 4.1 Implementasi Database



4.3.1 Simpanan

Tabel Simpanan dimana struktur tabel telah ditunjukkan pada **Tabel 3.6 Simpanan Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Simpanan pada *database* **Gambar 4.2 Implementasi Database Simpanan**

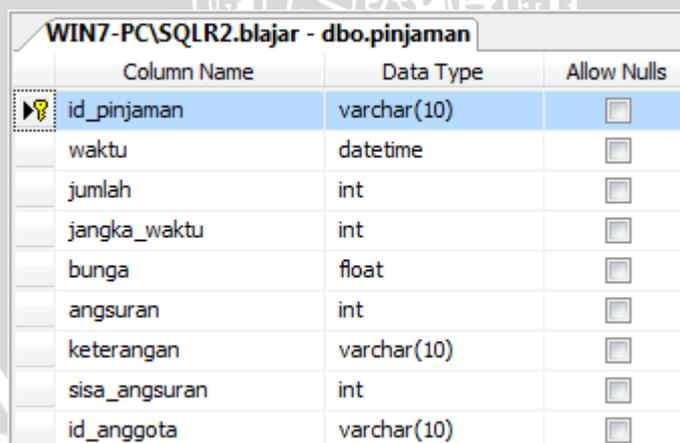


| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| id_simpanan | varchar(10) | <input type="checkbox"/> |
| waktu | datetime | <input type="checkbox"/> |
| jumlah | int | <input type="checkbox"/> |
| bunga | float | <input type="checkbox"/> |
| keterangan | varchar(10) | <input type="checkbox"/> |
| id_anggota | varchar(10) | <input type="checkbox"/> |

Gambar 4.2 Implementasi Database Simpanan

4.3.2 Pinjaman

Tabel Pinjaman dimana struktur tabel telah ditunjukkan pada **Tabel 3.7 Pinjaman Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Pinjaman pada *database* **Gambar 4.3 Implementasi Database Pinjaman**

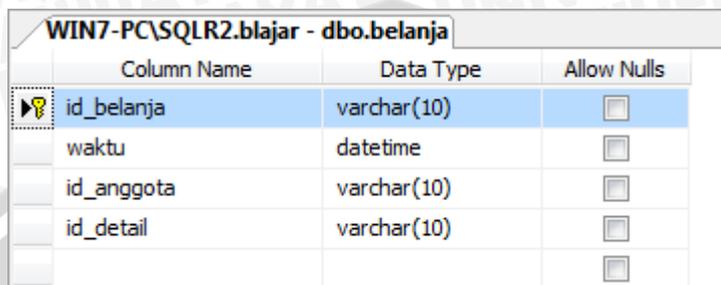


| Column Name | Data Type | Allow Nulls |
|--------------|-------------|--------------------------|
| id_pinjaman | varchar(10) | <input type="checkbox"/> |
| waktu | datetime | <input type="checkbox"/> |
| jumlah | int | <input type="checkbox"/> |
| jangka_waktu | int | <input type="checkbox"/> |
| bunga | float | <input type="checkbox"/> |
| angsuran | int | <input type="checkbox"/> |
| keterangan | varchar(10) | <input type="checkbox"/> |
| sisangsuran | int | <input type="checkbox"/> |
| id_anggota | varchar(10) | <input type="checkbox"/> |

Gambar 4.3 Implementasi Database Pinjaman

4.3.3 Belanja

Tabel Belanja dimana struktur tabel telah ditunjukkan pada **Tabel 3.8 Belanja Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Belanja pada *database* **Gambar 4.4 Implementasi Database Belanja**

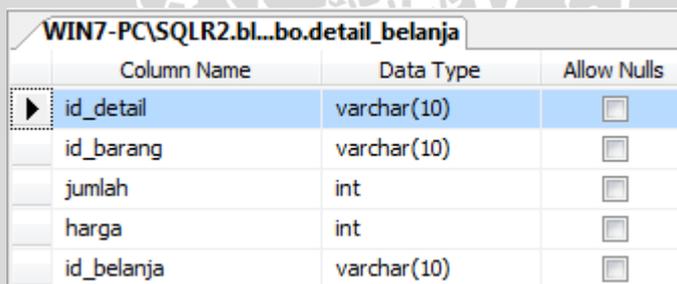


| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| id_belanja | varchar(10) | <input type="checkbox"/> |
| waktu | datetime | <input type="checkbox"/> |
| id_anggota | varchar(10) | <input type="checkbox"/> |
| id_detail | varchar(10) | <input type="checkbox"/> |
| | varchar(10) | <input type="checkbox"/> |

Gambar 4.4 Implementasi Database Belanja

4.3.4 Detail Belanja

Tabel Detail Belanja dimana struktur tabel telah ditunjukkan pada **Tabel 3.9 Detail Belanja Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Detail Belanja pada *database* **Gambar 4.5 Implementasi Database Detail Belanja**



| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| id_detail | varchar(10) | <input type="checkbox"/> |
| id_barang | varchar(10) | <input type="checkbox"/> |
| jumlah | int | <input type="checkbox"/> |
| harga | int | <input type="checkbox"/> |
| id_belanja | varchar(10) | <input type="checkbox"/> |

Gambar 4.5 Implementasi Database Detail Belanja

4.3.5 SHU

Tabel SHU dimana struktur tabel telah ditunjukkan pada **Tabel 3.11 SHU Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel SHU pada *database* **Gambar 4.6 Implementasi Database SHU**

| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| id_shu | varchar(10) | <input type="checkbox"/> |
| tahun | datetime | <input type="checkbox"/> |
| jumlah | int | <input type="checkbox"/> |
| id_anggota | varchar(10) | <input type="checkbox"/> |

Gambar 4.6 Implementasi Database SHU

4.3.6 Inbox SMS

Tabel Inbox SMS dimana struktur tabel telah ditunjukkan pada **Tabel 3.12 Inbox SMS Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Inbox SMS pada *database* **Gambar 4.7 Implementasi Database Inbox SMS**

| Column Name | Data Type | Allow Nulls |
|---------------|--------------|--------------------------|
| id_receive | int | <input type="checkbox"/> |
| no_pengirim | varchar(15) | <input type="checkbox"/> |
| isi_sms | varchar(500) | <input type="checkbox"/> |
| waktu_receive | datetime | <input type="checkbox"/> |

Gambar 4.7 Implementasi Database Inbox SMS

4.3.7 Sent SMS

Tabel Sent SMS dimana struktur tabel telah ditunjukkan pada **Tabel 3.13 Sent SMS Data Entity**, implementasi basis data direalisasikan dalam bentuk Tabel Sent SMS pada *database* **Gambar 4.8 Implementasi Database Sent SMS**

| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| id_sent | varchar(10) | <input type="checkbox"/> |
| id_anggota | varchar(10) | <input type="checkbox"/> |
| no_tujuan | varchar(15) | <input type="checkbox"/> |
| waktu_sent | datetime | <input type="checkbox"/> |
| isi_sms | text | <input type="checkbox"/> |

Gambar 4.8 Implementasi Database Sent SMS

4.4 Implementasi Method

Setiap *method* yang dimiliki *Use Case Diagram* pada **Gambar 3.7 Use Case Diagram** akan diimplementasikan sesuai dengan *use case*-nya. *Method* yang terdapat dalam Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS yaitu *method* untuk menambah data, mengedit data, menghapus data, *import* data excel, menerima SMS dan mengirimkan SMS.

Dalam laporan ini dicantumkan implementasi *method* menerima SMS dan mengirimkan SMS pada Form Receive SMS dan Form Kirim SMS Menu.

4.4.1 Terima SMS Anggota dan Kirim SMS Data Keuangan Anggota

Use Case terima SMS anggota dan kirim SMS data keuangan anggota pada **Gambar 3.7 Use Case Diagram** terdapat *Method* TampilSms() untuk menampilkan semua data pada Tabel inbox_sms, SmsMasuk() untuk menerima *request* SMS, ReceiveSms() untuk membalas *request* SMS, HapusSmsMasuk() untuk menghapus SMS yang masuk, KirimSmsMenu() untuk mengirimkan SMS menu, KirimSmsSimpanan() untuk mengirimkan SMS data simpanan, KirimSmsPinjaman() untuk mengirimkan SMS data pinjaman, KirimSmsBelanja untuk mengirimkan SMS data belanja, KirimSmsShu() untuk mengirimkan SMS

data SHU, KirimSmsSaldo() untuk mengirim SMS data saldo, KirimSmsFormatBelanja() untuk mengirimkan SMS format belanja online, HapusSmsKeluar() untuk menghapus SMS yang telah terkirim.

```

108 public void TampilSms()
109 {
110     dt = Koneksi.getDATATable("exec tampil_sms");
111     kryptonDataGridViewReceived.DataSource = dt;
112     kryptonDataGridViewReceived.BindingContext = this.BindingContext;
113 }
114
115 private void smsMasuk()
116 {
117     try
118     {
119         int uCountSMS = sms.CountSMSmessages(this.port); //menghitung sms
120         if (uCountSMS > 0)
121         {
122             string strCommand = "AT+CMGL=\"ALL\"";
123             collectionPesan = sms.ReadSMS(this.port, strCommand);
124
125             foreach (KonfigPesan msg in collectionPesan) // jika ada sms dan kemudian membaca sms
126             {
127                 dt = Koneksi.getDATATable("select id_anggota,no_hp from anggota where no_hp = '"
128                     + msg.NoPengirim + "'");
129                 if (dt.Rows.Count > 0)
130                 {
131                     Koneksi.eksekusi("exec insert_sms '" + msg.NoPengirim + "',"
132                         + msg.IsiSms + "','" + msg.WaktuReceive + "'");
133                 }
134             }
135             strCommand = "AT+CMGD=1,4";
136             sms.DeleteMsg(this.port, strCommand);
137         }
138     }
139     catch (Exception ex)
140     {
141         MessageBox.Show(ex.Message);
142     }
143 }
144
145 private void kryptonButtonReceive_Click(object sender, EventArgs e)
146 {
147     if (kryptonDataGridViewReceived.CurrentRow != null) //jika ada anggota yang dipilih
148     {
149         string selectedId = (string)kryptonDataGridViewReceived.CurrentRow.Cells[0].Value.ToString();
150         string selectedNoHp = (string)kryptonDataGridViewReceived.CurrentRow.Cells[1].Value; //mencata
151         string selectedIsi = (string)kryptonDataGridViewReceived.CurrentRow.Cells[2].Value; //mencatat
152         string selectedWkt = (string)kryptonDataGridViewReceived.CurrentRow.Cells[3].Value.ToString();
153
154         string pesan = "ID SMS Masuk : " + selectedId.ToString() + "\n No Pengirim : "
155             + selectedNoHp.ToString() + "\n Isi SMS : " + selectedIsi.ToString()
156             + "\n Waktu SMS Masuk : " + selectedWkt.ToString();
157
158         if (selectedIsi.Substring(0, 1) == "K")
159         {
160             {
161                 FormKirimSmsMenu formKirimSmsMenu = new FormKirimSmsMenu(selectedNoHp);
162                 formKirimSmsMenu.Show();
163             }
164         }
165         else if (selectedIsi == "1")
166         {
167             FormKirimSmsSimpanan formKirimSmsSimpanan = new FormKirimSmsSimpanan(selectedNoHp);
168             formKirimSmsSimpanan.Show();
169         }
170         else if (selectedIsi == "2")
171         {
172             FormKirimSmsPinjaman formKirimSmsPinjaman = new FormKirimSmsPinjaman(selectedNoHp);
173             formKirimSmsPinjaman.Show();
174         }
175         else if (selectedIsi == "3")
176     }
177 }
178
179
180

```

```
181     {
182         FormKirimSmsBelanja formKirimSmsBelanja = new FormKirimSmsBelanja(selectedNoHp);
183         formKirimSmsBelanja.Show();
184     }
185     else if (selectedIsi == "4")
186     {
187         FormKirimSmsShu formKirimSmsShu = new FormKirimSmsShu(selectedNoHp);
188         formKirimSmsShu.Show();
189     }
190     else if (selectedIsi == "5")
191     {
192         FormKirimSmsSaldo formKirimSmsSaldo = new FormKirimSmsSaldo(selectedNoHp);
193         formKirimSmsSaldo.Show();
194     }
195     else if (selectedIsi == "6")
196     {
197         FormKirimSmsListBelanja formKirimSmsListBelanja = new FormKirimSmsListBelanja(selectedNoHp);
198         formKirimSmsListBelanja.Show();
199     }
200     else //idbarang jumlah
201     {
202         FormKirimSmsBelanja formKirimSmsBelanja = new FormKirimSmsBelanja(selectedNoHp);
203         formKirimSmsBelanja.Show();
204     }
205 }
206 }
207 private void kryptonButtonHapusSms_Click(object sender, EventArgs e)
208 {
209     {
210         if (kryptonDataGridViewReceived.CurrentRow != null) //jika ada barang yg dipilih
211         {
212             int selectedId = (int)kryptonDataGridViewReceived.CurrentRow.Cells[0].Value; //mencata
213
214             if (MessageBox.Show("Apakah ingin menghapus data SMS ini?", "Hapus Data SMS",
215                 MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes) //jika user
216             {
217                 Koneksi.eksekusi("delete from inbox_sms where id_receive = '" + selectedId + "'");
218                 TampilSms();
219             }
220         }
221     }
222 }
```

Gambar 4.9 Implementasi Method TampilSms(), SmsMasuk(), ReceiveSms(), HapusSms() pada Form Receive SMS

Method TampilSms() pada **Gambar 4.9 Implementasi Method TampilSms(),SmsMasuk(),ReceiveSms(), HapusSms() pada Form Receive SMS** baris 108-113 menampilkan semua data yang terdapat pada Tabel inbox_sms. Pada *Method* SmsMasuk() baris 115-143 menghitung jumlah SMS yang masuk pada modem kemudian menyeleksi apakah nomor pengirim adalah nomor anggota koperasi dan apakah ID anggota yang digunakan sesuai. Jika sesuai maka akan dimasukkan ke dalam *database* Tabel inbox_sms. Jika tidak sesuai maka akan terhapus dan tidak di-*input*-kan ke dalam *database*. Pada *Method* ReceiveSms() baris 150-206 membaca isi SMS dan mengarahkan ke *form* kirim SMS sesuai dengan isi SMS. Pada *Method* HapusBroadcast() baris 208-221 menghapus data SMS masuk sesuai dengan ID SMS masuk.

```
46 private void isiSms()
47 {
48     kryptonTextBoxIsiSms.Text = "1. Cek Simpanan" + "\r\n" + "2. Cek Pinjaman" + "\r\n"
49     + "3. Cek Belanja" + "\r\n" + "4. Cek SHU" + "\r\n" + "5. Cek Saldo" + "\r\n"
50     + "6. Belanja";
51 }
65 private void kryptonButtonKirimMenu_Click(object sender, EventArgs e)
66 {
67     try
68     {
69         if (sms.sendMsg(Koneksi.portku, this.kryptonTextBoxNoHp.Text, this.kryptonTextBoxIsiSms.Text))
70         {
71             MessageBox.Show("SMS Terkirim", "Status", MessageBoxButtons.OK, MessageBoxIcon.Information);
72         }
73
74         Koneksi.eksekusi("insert into sent_sms values ('" + kryptonTextBoxIdSentSms.Text + "','" +
75         + kryptonTextBoxIdAnggota.Text + "','" + kryptonTextBoxNoHp.Text
76         + "',getdate(),' + kryptonTextBoxIsiSms.Text + '");
77     }
78     catch (Exception ex)
79     {
80         MessageBox.Show(ex.Message);
81     }
82 }
```

Gambar 4.10 Implementasi Method KirimSmsMenu() pada Form Kirim SMS Menu

Method KirimSmsMenu() pada **Gambar 4.10 Implementasi Method KirimSmsMenu() pada Form Kirim SMS Menu** baris 65-82 mengirimkan (membalas) SMS menu-menu yang dapat pilih oleh *user* (Anggota Koperasi Karyawan Sejahtera PT. Multibreeder Adirama Indonesia Tbk.) dan kemudian menyimpan di dalam *database* Tabel sent_sms. Baris 74-76 merupakan perintah *sql* untuk memasukkan data SMS terkirim ke dalam tabel.

4.5 Implementasi Antarmuka

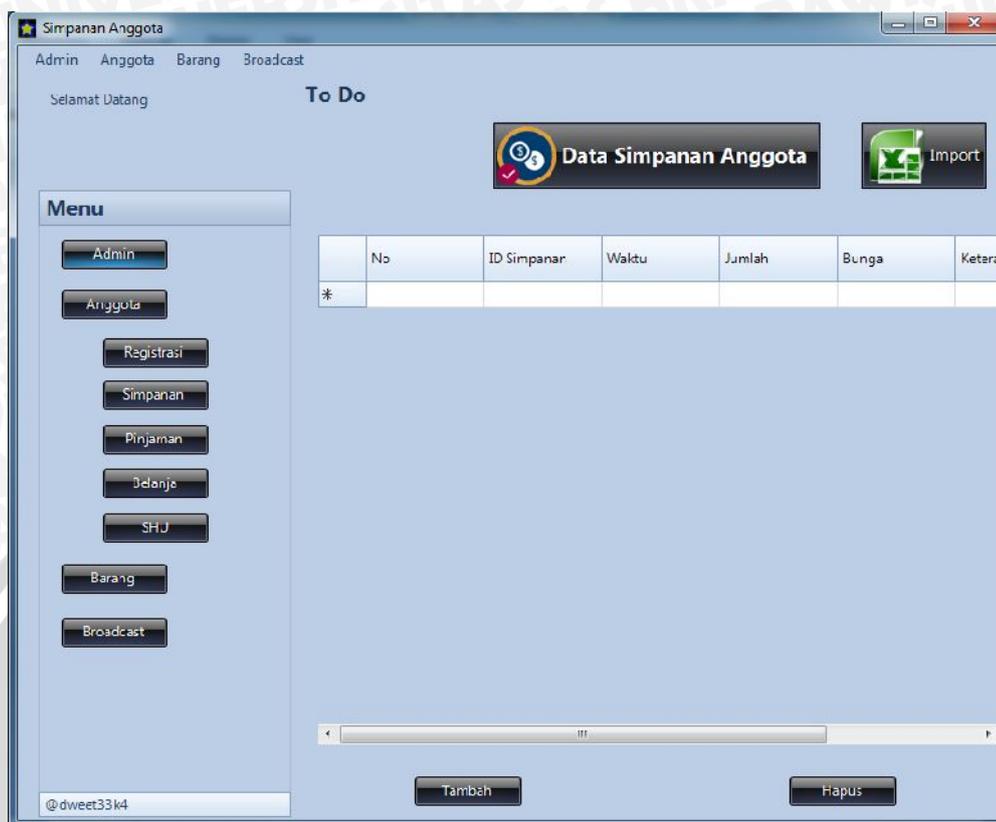
Implementasi antarmuka perangkat lunak Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS berdasarkan perancangan terdiri dari implementasi antarmuka Form Login, implementasi antarmuka Form Koneksi, implementasi antarmuka Form Admin, implementasi antarmuka Form Input Data Admin, implementasi antarmuka Form Registrasi Anggota, implementasi antarmuka Form Input Data Registrasi, implementasi antarmuka Form Simpanan, implementasi antarmuka Form Input Data Simpanan, implementasi antarmuka Form Pinjaman, implementasi antarmuka Form Input Data Pinjaman, implementasi antarmuka Form Edit Sisa Angsuran, implementasi antarmuka Form Belanja, implementasi antarmuka Form Input Data Belanja, implementasi

antarmuka Form Detail Belanja, implementasi antarmuka Form Input Data Detail Belanja, implementasi antarmuka Form SHU, implementasi antarmuka Form Input Data SHU, implementasi antarmuka Form Barang, implementasi antarmuka Form Input Data Barang, implementasi antarmuka Form Broadcast, implementasi antarmuka Form Input Data Broadcast, implementasi antarmuka Form SMS Masuk, implementasi Form Kirim SMS Menu, implementasi Form Kirim SMS Simpanan, implementasi Form Kirim SMS Pinjaman, implementasi Form Kirim SMS Belanja, implementasi Form Kirim SMS SHU, implementasi Form Kirim SMS Saldo, dan implementasi antarmuka Form SMS Keluar.

Pada laporan ini dijelaskan mengenai implementasi antarmuka Form Simpanan, Form Pinjaman, Form Belanja, Form SHU, dan Form Kirim SMS.

4.5.1 Implementasi Antarmuka Form Simpanan Anggota dan Implementasi Antarmuka Form Input Data Simpanan

Implementasi antarmuka Form Simpanan Anggota merupakan *form* yang digunakan untuk menampilkan data yang terdapat pada *database* Tabel Simpanan. *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka. Terdapat satu tombol untuk menambah data isian, dan satu tombol untuk menghapus data. *Form* isian untuk mengisikan data simpanan yaitu jumlah simpanan, keterangan simpanan, dan ID Anggota. **Gambar 4.11 Implementasi Antarmuka Form Simpanan Anggota** menunjukkan tampilan implementasi antarmuka Form Simpanan Anggota dan **Gambar 4.12 Implementasi Antarmuka Form Input Data Simpanan** menunjukkan tampilan implementasi antarmuka Form Input Data Simpanan.



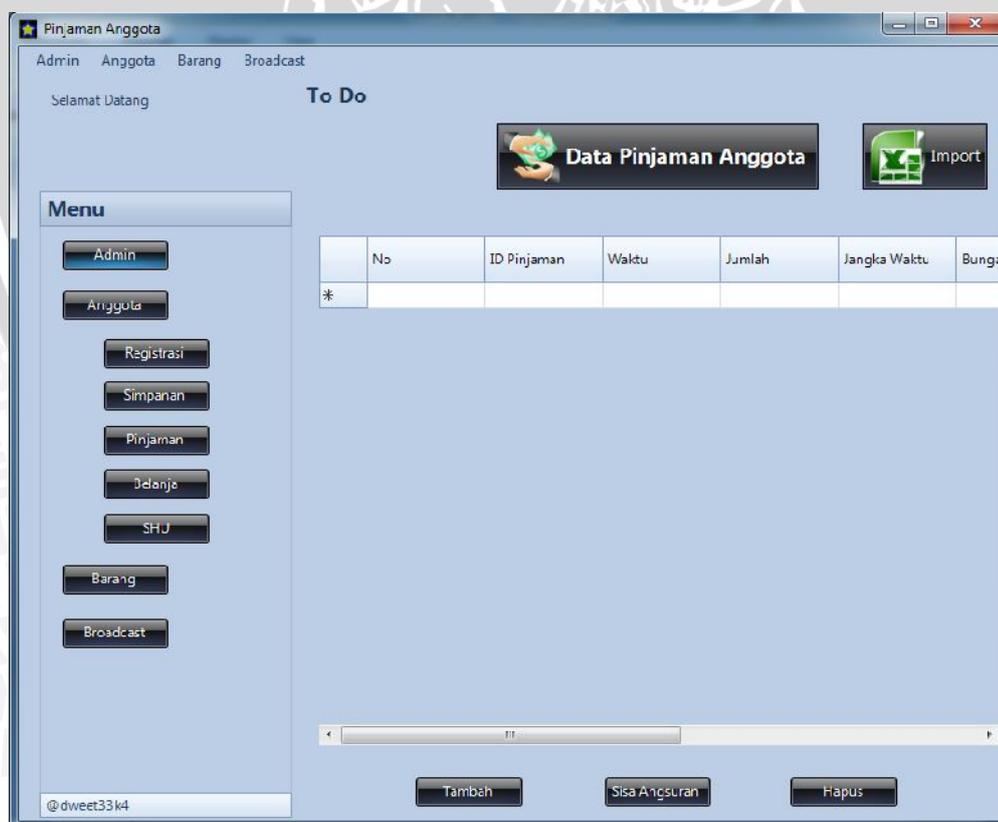
Gambar 4.11 Implementasi Antarmuka Form Simpanan Anggota



Gambar 4.12 Implementasi Antarmuka Form Input Data Simpanan

4.5.2 Implementasi Antarmuka Form Pinjaman Anggota dan Implementasi Antarmuka Form Input Data Pinjaman

Implementasi antarmuka Form Pinjaman Anggota merupakan *form* yang digunakan untuk menampilkan data yang terdapat pada *database* Tabel Pinjaman. *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka. Terdapat satu tombol untuk menambah data isian, satu tombol untuk mengubah data (hanya mengubah sisa angsuran), dan satu tombol untuk menghapus data. *Form* isian untuk mengisi data pinjaman yaitu jumlah pinjaman, jangka waktu, bunga, angsuran, keterangan pinjaman, sisa angsuran, dan ID Anggota. **Gambar 4.13 Implementasi Antarmuka Form Pinjaman Anggota** menunjukkan tampilan implementasi antarmuka Form Pinjaman Anggota, **Gambar 4.14 Implementasi Antarmuka Form Input Data Pinjaman** menunjukkan tampilan implementasi antarmuka Form Input Data Pinjaman dan **Gambar 4.15 Implementasi Antarmuka Form Edit Sisa Angsuran** menunjukkan tampilan implementasi antarmuka Form Edit Sisa Angsuran.



Gambar 4.13 Implementasi Antarmuka Form Pinjaman Anggota



Tambah Data Pinjaman Anggota

Input Data Pinjaman Anggota

ID Pinjaman

Jumlah Rp

Jangka Waktu Bulan

Bunga (%)

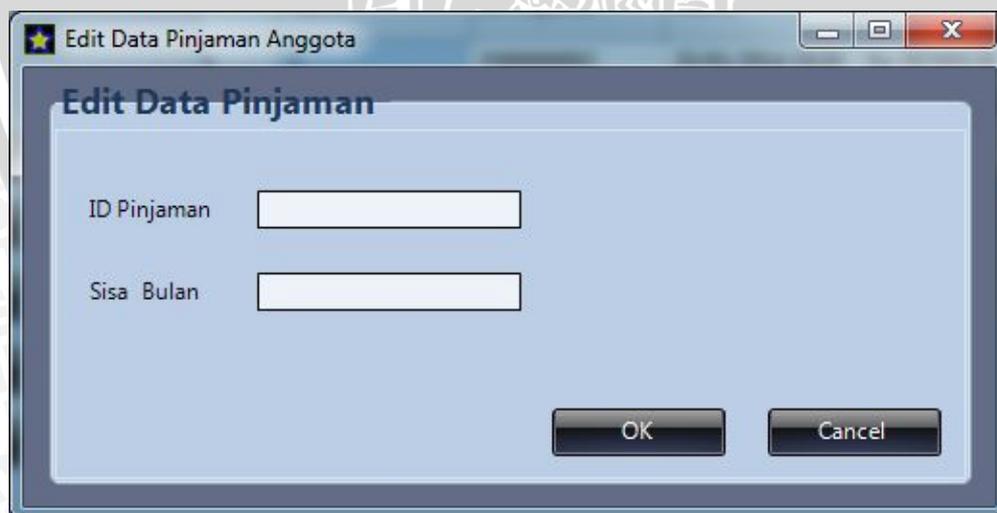
Angsuran (x) Sisa (x)

Keterangan

ID Anggota

OK Cancel

Gambar 4.14 Implementasi Antarmuka Form Input Data Pinjaman



Edit Data Pinjaman Anggota

Edit Data Pinjaman

ID Pinjaman

Sisa Bulan

OK Cancel

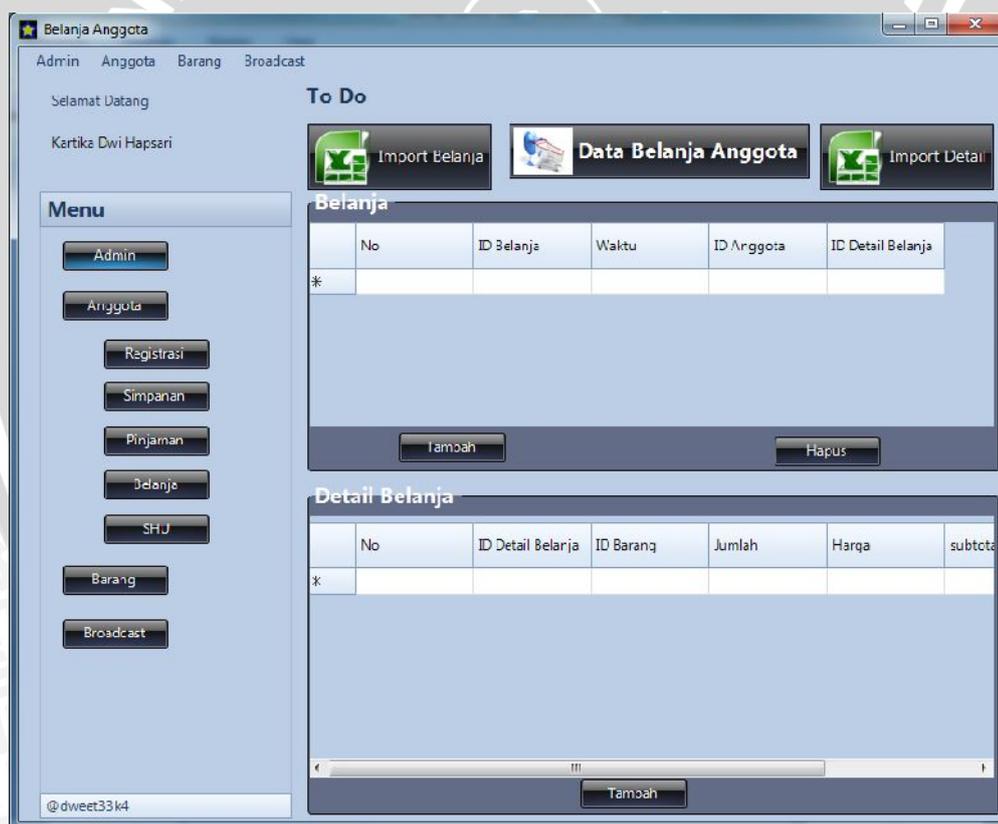
Gambar 4.15 Implementasi Antarmuka Form Edit Sisa Angsuran

4.5.3 Implementasi Antarmuka Form Belanja Anggota dan Implementasi Antarmuka Form Input Data Belanja

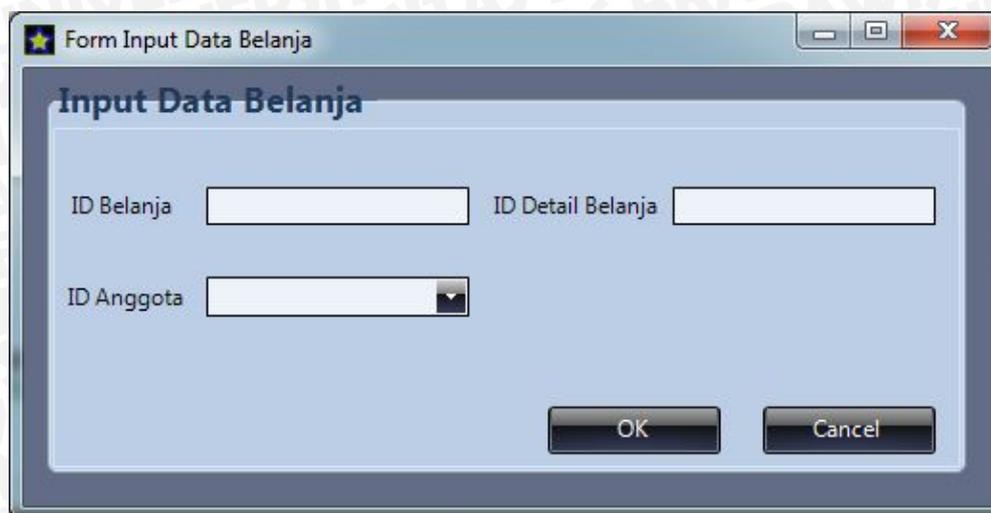
Implementasi antarmuka Form Belanja Anggota merupakan *form* yang digunakan untuk menampilkan data yang terdapat pada *database* Tabel Belanja. *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka. Terdapat satu tombol untuk menambah data isian, dan satu tombol untuk menghapus data.

Form isian untuk mengisi data belanja yaitu ID Anggota. **Gambar 4.16**

Implementasi Antarmuka Form Belanja Anggota menunjukkan tampilan implementasi antarmuka Form Belanja Anggota dan **Gambar 4.17 Implementasi Antarmuka Form Input Data Belanja** menunjukkan tampilan implementasi antarmuka Form Input Data Belanja.



Gambar 4.16 Implementasi Antarmuka Form Belanja Anggota



The image shows a screenshot of a web application window titled "Form Input Data Belanja". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light blue and contains three input fields: "ID Belanja" (a text box), "ID Detail Belanja" (a text box), and "ID Anggota" (a dropdown menu). At the bottom right of the form area, there are two buttons: "OK" and "Cancel".

Gambar 4.17 Implementasi Antarmuka Form Input Data Belanja

4.5.4 Implementasi Antarmuka Form Detail Belanja Anggota dan Implementasi Antarmuka Form Input Data Detail Belanja

Implementasi antarmuka Form Detail Belanja Anggota merupakan *form* yang digunakan untuk menampilkan data yang terdapat pada *database* Tabel Detail Belanja. *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka. Terdapat satu tombol untuk menambah data isian, dan satu tombol untuk menghapus data. *Form* isian untuk mengisi data detail belanja yaitu ID Barang dan jumlah barang. **Gambar 4.18 Implementasi Antarmuka Form Input Data Detail Belanja** menunjukkan tampilan implementasi antarmuka Form Input Data Detail Belanja.

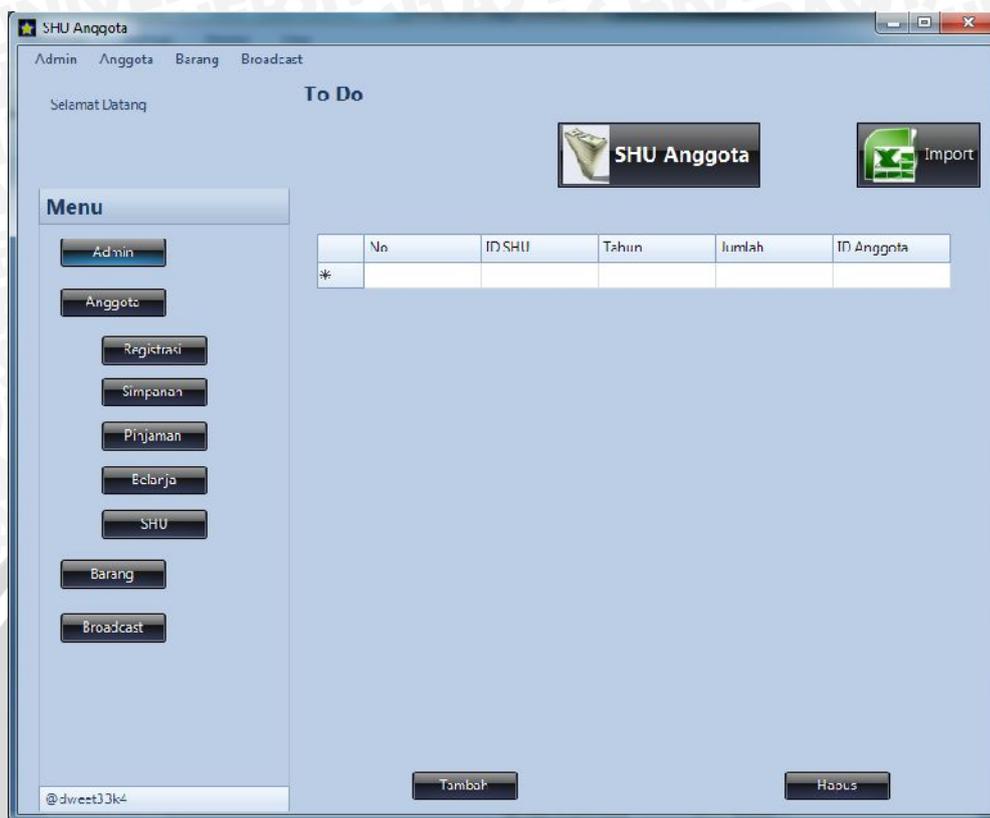


The image shows a screenshot of a software application window titled "Tambah Data Detail Belanja Anggota". Inside the window, there is a form titled "Input Data Detail Belanja". The form contains six input fields arranged in two columns. The left column has "ID Detail Belanja", "ID Barang", and "Jumlah". The right column has "Harga", "ID Anggota", and "ID Belanja". At the bottom of the form, there are two buttons: "OK" and "Cancel".

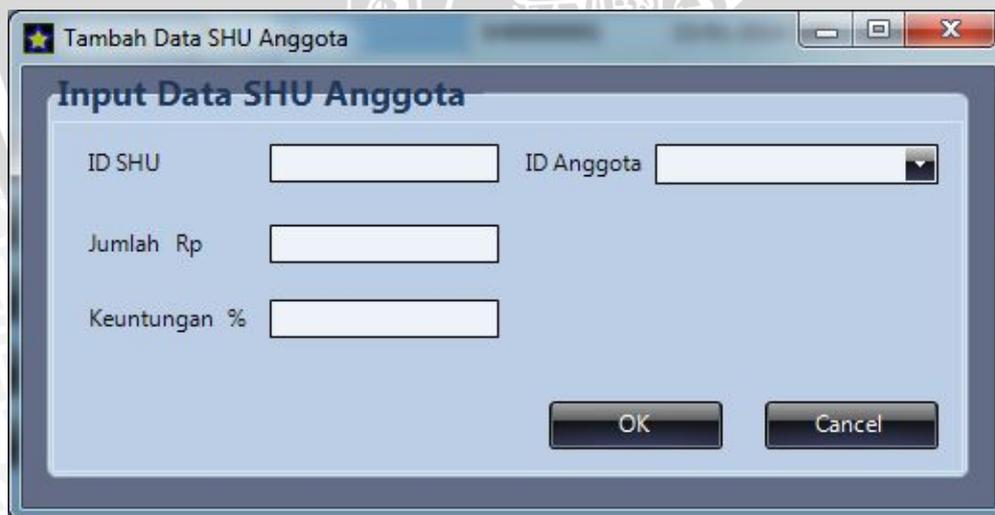
Gambar 4.18 Implementasi Antarmuka Form Input Data Detail Belanja

4.5.5 Implementasi Antarmuka Form SHU Anggota dan Implementasi Antarmuka Form Input Data SHU

Implementasi antarmuka Form SHU Anggota merupakan *form* yang digunakan untuk menampilkan data yang terdapat pada *database* Tabel SHU. *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka. Terdapat satu tombol untuk menambah data isian, dan satu tombol untuk menghapus data. *Form* isian untuk mengisi data shu yaitu jumlah perolehan shu, keuntungan dan ID Anggota. **Gambar 4.19 Implementasi Antarmuka Form SHU Anggota** menunjukkan tampilan implementasi antarmuka Form SHU Anggota dan **Gambar 4.20 Implementasi Antarmuka Form Input Data SHU** menunjukkan tampilan implementasi antarmuka Form Input Data SHU.



Gambar 4.19 Implementasi Antarmuka Form SHU Anggota



Gambar 4.20 Implementasi Antarmuka Form Input Data SHU

4.5.6 Implementasi Antarmuka Form Kirim SMS

Implementasi antarmuka Form Kirim SMS merupakan *form* yang digunakan untuk isian balasan SMS sesuai dengan *request* dari *user* (Anggota Koperasi Karyawan Sejahtera PT. Multibreeder Adirama Indonesia Tbk.). *Form* ini dibuat berdasarkan implementasi dari perancangan antarmuka **Gambar 4.21 Implementasi Antarmuka Form Kirim SMS Saldo** menunjukkan tampilan implementasi antarmuka Form Kirim SMS Saldo.



Gambar 4.21 Implementasi Antarmuka Form Kirim SMS Saldo

4.6 Implementasi Manajemen Resiko

Implementasi manajemen resiko pada **Gambar 3.11 Manajemen Resiko** perangkat lunak Aplikasi Layanan Pengecekan Tabungan dan Belanja Online Berbasis SMS berdasarkan perancangan terdiri dari tiga antisipasi dan adaptasi resiko. Yaitu, sosialisasi melalui lisan dan tulisan (tutorial untuk admin dan tutorial untuk pengguna dalam hal ini adalah anggota koperasi), meng-*import*-kan data dari Excel yang kemudian langsung tersimpan ke dalam *database*, dan tidak adanya fasilitas untuk mengganti data keuangan.

4.6.1 Sosialisasi Melalui Lisan dan Tulisan

Melalui tutorial penggunaan aplikasi, baik untuk bagian admin maupun bagian *user* (bagian SMS) dapat menjadi solusi untuk resiko sosialisasi tidak merata.

4.6.2 Import Data File Excel

Melalui fungsi *import* yang disediakan oleh aplikasi, dapat menjadi solusi untuk resiko membutuhkan waktu yang lama dalam pemindahan data.

4.6.3 Tidak ada Fasilitas untuk Mengganti Data Keuangan

Dengan tidak adanya fasilitas “Edit” untuk data keuangan, dapat menjadi solusi untuk resiko data tidak akurat atau tidak valid.

