

## BAB IV HASIL DAN PEMBAHASAN

### 4.1. Perancangan Sistem

Proses pencatatan terhadap transaksi tabungan siswa saat ini masih dilakukan dengan cara tradisional yaitu manual dan memerlukan interaksi antara wali siswa dan wali kelas. Guru kelas melanjutkan ke langkah selanjutnya untuk petugas atau pengelola tabungan, menulis transaksi untuk buku tabungan siswa dan mencatatnya di buku besar pengelola tabungan untuk memverifikasi transaksi yang telah selesai.

Permasalahan yang ada saat ini adalah sistem yang berjalan dalam pengelolaan tabungan siswa-siswi di TK Ar-Rasyid masih dilakukan secara konvensional atau manual. Pencatatan dan penyimpanan data ditulis pada buku kas yang dimiliki oleh wali kelas dan petugas serta cara perhitungan dilakukan dengan cara perhitungan manual atau menggunakan kalkulator dan hal lainnya setiap data yang disimpan berbentuk arsip-arsip, sehingga besar kemungkinan arsip tersebut untuk dapat hilang, catatan rusak karena terkena air, terbakar, dan lain-lain. Oleh sebab itu, dibutuhkan sistem informasi pengelolaan data tabungan siswa-siswi, yang dapat memudahkan petugas dalam pengelolaan data serta perhitungan jumlah tabungan serta pendataan data tabungan yang akurat.

### 4.2. Metode Prototyping

Peneliti dalam tahapan pengembangan metode prototyping dengan langkah-langkah sebagai berikut:

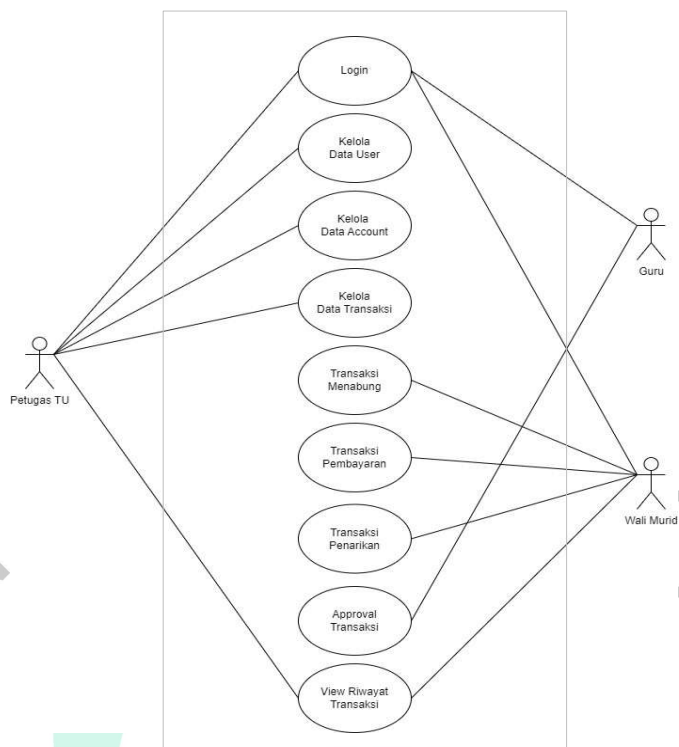
- Analisis; melakukan Analisis Kebutuhan.
- Membangun Prototipe; membuat simulasi input – output model prototipe.
- Evaluasi Prototipe; melakukan *review* pada hasil simulasi prototipe dengan memeriksa apakah prototipe yang dibangun sudah sesuai dengan Requirements.
- Pemrograman Sistem; memilih bahasa pemrograman yang akan digunakan dalam membangun aplikasi.
- Menguji Sistem; melakukan pengujian dan evaluasi aplikasi.

### 4.3. Perancangan Diagram Sistem Usulan

Untuk perancangan aplikasi, desain yang digunakan penelitian ini yaitu *Unified Modelling Language (UML)*. Hal ini dilakukan untuk memudahkan perancangan aplikasi pendataan tabungan siswa serta penggunaan UML digunakan lebih tepat dalam perancangan aplikasi bersifat *object oriented*.

#### 4.3.1. Use-case Diagram

Untuk menggambarkan hubungan antara aktor yang ada di aplikasi yang akan dibuat dengan sistem, dimana aktor terdiri dari 3 aktor yaitu Petugas TU, Guru dan Wali murid sesuai dengan analisis kebutuhan sistem, dapat dilihat pada gambar dibawah ini.



**Gambar 1- 9.** Use-case Diagram Aplikasi Mobile Tabungan Siswa

Pada aplikasi pendataan tabungan terdapat 3 aktor yaitu Petugas TU, Guru dan Wali Murid Berikut penjelasan dan interaksi yang terjadi antara aktor dan sistem sebagai berikut:

- **Petugas TU**  
Aktor petugas TU merupakan pengguna yang bertanggung jawab dalam mengelola yang berkaitan dengan pendataan tabungan. Dimana petugas memiliki tugas dapat melakukan pengelolaan pengguna, pengelolaan data pengguna atau siswa-siswi, pengelolaan transaksi dan pengelolaan pendaftaran tabungan serta mengelola data pemilik tabungan.
- **Guru**  
Aktor guru pengguna yang melakukan konfirmasi isi saldo, tarik saldo, status pembayaran serta dapat melihat list data saldo seluruh pemilik tabungan.
- **Wali Murid**  
Aktor Wali murid merupakan pengguna yang dapat melakukan dengan kegiatan isi saldo, tarik saldo, melakukan transaksi pembayaran dan melihat informasi tabungan.

#### 4.3.2. Spesifikasi Use-case Diagram

Spesifikasi *usecase* diagram adalah penulis menjelaskan secara detail mengenai gambar yang terdapat pada diagram *usecase*. Berikut ini Spesifikasi Usecase Diagram:

**Tabel 1- 9.** Spesifikasi Use-case Login

Use Case	Login
----------	-------

<i>Actor</i>	Petugas TU, Guru, Wali Murid
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengelola Data User
<i>Pre-conditions</i>	Pengguna sudah terdaftar dalam aplikasi pendataan tabungan tetapi belum masuk sisem.
<i>Post-conditions</i>	Masuk ke aplikasi pendataan pada halaman utama masing-masing aktor
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi pendataan tabungan.</li> <li>2. Memasukan pengguna sebagai email pengguna.</li> <li>3. Memasukkan kata sandi.</li> <li>4. Sistem melakukan validasi akun pengguna yang dimasukkan, dan mengarahkan pada setiap halaman utama sesuai dengan level pengguna.</li> <li>5. Jika pengguna adalah admin maka akan masuk pada halaman admin.</li> <li>6. Jika pengguna adalah guru akan masuk pada halaman guru.</li> <li>7. Jika pengguna adalah siswa/siswi, maka akan masuk pada halaman user siswa/siswa.</li> </ol>
<i>Extensions / Alternate Scenario</i>	<ol style="list-style-type: none"> <li>1. Ketika akun pengguna tidak ditemukan, maka sistem mengeluarkan pesan kesalahan.</li> <li>2. Pengguna tetap berada di halaman login</li> </ol>
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 10.** Spesifikasi Use-case Cek Total Saldo (Wali Murid)

Use Case	Melihat Jumlah Saldo
<i>Actor</i>	Wali Murid
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Melihat Jumlah Saldo
<i>Pre-conditions</i>	Total saldo terhitung secara akumulatif.
<i>Post-conditions</i>	Masuk ke halaman menu kemudian terdapat tampilan Total Saldo.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi dan terdapat tampilan jumlah total saldo.</li> </ol>
<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 11.** Spesifikasi Use-case Cek Riwayat Transaksi (Wali Murid)

Use Case	View Riwayat Transaksi
<i>Actor</i>	Wali Murid
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Melihat Data Transaksi.

<i>Pre-conditions</i>	Melihat list data transaksi yang telah dilakukan.
<i>Post-conditions</i>	Terdapat button/tombol yang dapat menampilkan data history transaksi.
<i>Main Success Scenario</i>	1. Mengakses aplikasi dan terdapat tampilan list data dalam tampilan data table.
<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 12.** Spesifikasi Use-case Isi Saldo (Wali Murid)

Use Case	Isi Saldo
<i>Actor</i>	Wali Murid
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Melakukan Isi Saldo.
<i>Pre-conditions</i>	Tampil form untuk mengisi saldo, terdapat pilhan metode transfer/tunai dan terdapat upload bukti kirim berupa photo serta input keterangan.
<i>Post-conditions</i>	Klik button atau tombol Isi Saldo lalu tampil form isi saldo, input form, upload phoho, bukti transaksi lalu Confirm.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Mengakses form isi saldo.</li> <li>2. Input jumlah saldo yang akan ditambahkan, pilih metode transfer/tunai, input info pengirim, input pengiriman kemana, upload photo bukti kirim dan input keterangan.</li> <li>3. Confirm.</li> </ol>
<i>Extensions / Alternate Scenario</i>	<ol style="list-style-type: none"> <li>1. Form isi saldo yang telah confirm akan muncul pada halaman history dengan status Dalam Proses.</li> <li>2. Status Isi Saldo akan berubah berhasil setelah Guru melakukan konfirmasi.</li> <li>3. Jumlah saldo akan bertambah sesuai dengan pengisian saldo.</li> </ol>
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 13.** Spesifikasi Use-case Tarik Saldo (Wali Murid)

Use Case	Tarik Saldo
<i>Actor</i>	Wali Murid
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Melakukan Penarikan Saldo.
<i>Pre-conditions</i>	Tampil form untuk penarikan saldo, terdapat pilhan metode transfer/tunai dan terdapat upload bukti kirim berupa photo serta input keterangan.

<b>Post-conditions</b>	Klik button atau tombol Tarik Saldo lalu tampil form tarik saldo, input form, pilihan metode cara penerimaan, rekening penerima, input jumlah input keterangan lalu Confirm.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Mengakses form tarik saldo.</li> <li>2. Input jumlah saldo yang akan ditarik, pilih metode penerimaan transfer/tunai, input info rekening penerima, input jumlah penarikan saldo dan input keterangan.</li> <li>3. Confirm.</li> <li>4.</li> </ol>
<b>Extensions / Alternate Scenario</b>	<ol style="list-style-type: none"> <li>1. Form Tarik saldo yang telah confirm akan muncul pada halaman history dengan status Dalam Proses.</li> <li>2. Status Isi Saldo akan berubah berhasil setelah Guru melakukan konfirmasi.</li> <li>3. Jumlah Saldo akan berkurang sesuai dengan jumlah penarikan.</li> </ol>
<b>Priority</b>	High
<b>Special Requirements</b>	-
<b>Open Questions</b>	-

**Tabel 1- 14.** Spesifikasi Use-case Pembayaran (Wali Murid)

Use Case	Pembayaran
<b>Actor</b>	Wali Murid
<b>Stakeholders and Interest</b>	-
<b>Trigger</b>	Melakukan SPP/Ektra/Transport/Seragam/Buku. Pembayaran
<b>Pre-conditions</b>	Tampil form pembayaran.
<b>Post-conditions</b>	Klik button atau tombol Pembayaran sesuai nama label SPP/Ektra/Buku/Trnasport/Seragam lalu tampil form pembayaran, input form, input jumlah pembayaran, input keterangan lalu Confirm.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Mengakses form pembayaran.</li> <li>2. Input jumlah pembayaran dan input keterangan.</li> <li>3. Confirm.</li> <li>4.</li> </ol>
<b>Extensions / Alternate Scenario</b>	<ol style="list-style-type: none"> <li>1. Form Pembayaran yang telah confirm akan muncul pada halaman history dengan status Dalam Proses.</li> <li>2. Status Pembayaran akan berubah berhasil setelah Guru melakukan konfirmasi.</li> <li>3. Jumlah saldo akan berkurang sesuai dengan jumlah pembayaran.</li> </ol>
<b>Priority</b>	High
<b>Special Requirements</b>	-
<b>Open Questions</b>	-

**Tabel 1- 15.** Spesifikasi Use-case Cek Total Saldo (Petugas TU)

Use Case	View Total Saldo
<i>Actor</i>	Petugas TU
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Melihat Total Saldo
<i>Pre-conditions</i>	Total saldo siswa terhitung secara akumulatif.
<i>Post-conditions</i>	Masuk ke halaman menu data pengguna, kemudian terdapat tampilan Total Saldo Siswa. 4. Mengakses aplikasi dan terdapat tampilan jumlah total saldo siswa.
<i>Main Success Scenario</i>	
<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 16.** Spesifikasi Use-case Kelola Data Siswa (Petugas TU)

Use Case	Kelola Data User
<i>Actor</i>	Petugas TU
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses data sekolah dari siswa/siswi, guru serta petugas TU.
<i>Pre-conditions</i>	Proses memasukan data siswa-siswi, guru dan petugas yang aktif disekolah.
<i>Post-conditions</i>	Klik button Data User akan masuk ke halaman data siswa, kemudian melakukan pengelolaan data siswa, seperti menambahkan data, mengubah, bahkan menghapus data siswa.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi data sekolah.</li> <li>2. Melakukan penambahan data siswa/siswi, guru dan petugas.</li> <li>3. Mengubah atau mengedit data.</li> <li>4.</li> <li>5. Sistem akan menampilkan pop-up apakah data pengguna yang diubah akan diedit, hapus atau delete.</li> </ol>
<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 17.** Spesifikasi Use-case Kelola Account (Petugas TU)

Use Case	Kelola Account
<i>Actor</i>	Petugas TU
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses account pemilik tabungan

<b>Pre-conditions</b>	Siswa yang terdaftar pada tabungan siswa tampil dalam list data pemilik tabungan.
<b>Post-conditions</b>	Klik button Account, masuk kedalam list data account kemudian melakukan pengelolaan data account, seperti menambahkan, mengubah, bahkan menghapus data pemilik tabungan.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi pendataan data pemilik tabungan.</li> <li>2. Melakukan penambahan data pemilik tabungan.</li> <li>3. Mengubah data pemilik tabungan.</li> <li>4. Sistem akan menampilkan pop-up apakah data pengguna yang diubah akan diedit, hapus atau delete.</li> </ol>
<b>Extensions / Alternate Scenario</b>	-
<b>Priority</b>	High
<b>Special Requirements</b>	-
<b>Open Questions</b>	-

**Tabel 1- 18.** Spesifikasi Use-case Kelola Data Transaksi (Petugas TU)

Use Case	Kelola Data Transaksi
<b>Actor</b>	Petugas TU
<b>Stakeholders and Interest</b>	-
<b>Trigger</b>	Mengakses data transaksi, terjadi human error.
<b>Pre-conditions</b>	Setiap data transaksi tampil pada halaman list data transaksi.
<b>Post-conditions</b>	Klik button Transaksi, tampil halaman data transi, kemudian melakukan pengelolaan data transaksi, seperti menambahkan data, mengubah, bahkan menghapus data pengguna tabungan.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi pendataan transaksi tabungan siswa-siswi.</li> <li>2. Melakukan penambahan data transaksi.</li> <li>3. Mengubah data transaksi.</li> <li>4. Sistem akan menampilkan pop-up apakah data pengguna yang diubah akan disimpan atau dibatalkan.</li> <li>5. Menghapus data transaksi.</li> </ol>
<b>Extensions / Alternate Scenario</b>	-
<b>Priority</b>	High
<b>Special Requirements</b>	-
<b>Open Questions</b>	-

**Tabel 1- 19.** Spesifikasi Use-case View Total Saldo (Guru)

Use Case	View Total Saldo
<b>Actor</b>	Guru
<b>Stakeholders and Interest</b>	-
<b>Trigger</b>	Melihat Total Saldo
<b>Pre-conditions</b>	Total saldo siswa terhitung secara akumulatif.

<i>Post-conditions</i>	Masuk ke halaman menu data pengguna, kemudian terdapat tampilan Total Saldo Siswa.
	Mengakses aplikasi dan terdapat tampilan jumlah total saldo siswa.

*Main Success Scenario*

<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	<i>High</i>
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 20.** Spesifikasi Use-case Konfirmasi Isi Saldo (Guru)

Use Case	Konfirmasi Isi Saldo
<i>Actor</i>	Guru
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses list data transaksi.
<i>Pre-conditions</i>	Konfirmasi status isi saldo.
<i>Post-conditions</i>	Guru memeriksa jumlah isi saldo dan dana yang diterima, memeriksa bukti pengiriman jika melalui transfer lalu klik button berkedip Menunggu Proses.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>Status button Menunggu Proses akan berubah menjadi Selesai.</li> <li>Tampilan popup alert untuk melanjutkan proses Selesai.</li> </ol>
<i>Extensions / Alternate Scenario</i>	Konfirmasi akan mengubah data history status pada tampilan status transaksi siswa menjadi Berhasil.
<i>Priority</i>	<i>High</i>
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 21.** Spesifikasi Use-case Konfirmasi Tarik Saldo (Guru)

Use Case	Konfirmasi Tarik Saldo
<i>Actor</i>	Guru
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses list data transaksi.
<i>Pre-conditions</i>	Konfirmasi status tarik saldo.
<i>Post-conditions</i>	Guru memeriksa jumlah isi saldo dan jumlah saldo yang akan ditarik tidak melebihi dari jumlah isi saldo siswa lalu klik button berkedip Menunggu Proses.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>Status button Menunggu Proses akan berubah menjadi Selesai.</li> <li>Tampilan popup alert untuk melanjutkan proses Selesai.</li> </ol>
<i>Extensions / Alternate Scenario</i>	Konfirmasi akan mengubah data history status pada tampilan status transaksi siswa menjadi Berhasil.
<i>Priority</i>	<i>High</i>
<i>Special Requirements</i>	-
<i>Open Questions</i>	-



**Tabel 1- 22.** Spesifikasi Use-case Konfirmasi Pembayaran (Guru)

Use Case	Konfirmasi Pembayaran
<i>Actor</i>	Guru
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses list data transaksi.
<i>Pre-conditions</i>	Konfirmasi status pembayaran.
<i>Post-conditions</i>	Guru memeriksa transaksi, tipe pembayaran dan pembayaran serta jumlah pembayaran tidak melebihi dari jumlah isi saldo siswa lalu klik button berkedip Menunggu Proses.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Status button Menunggu Proses akan berubah menjadi Selesai.</li> <li>2. Tampilan popup alert untuk melanjutkan proses Selesai.</li> </ol>
<i>Extensions / Alternate Scenario</i>	Konfirmasi akan mengubah data history status pada tampilan status transaksi siswa menjadi Berhasil.
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 23.** Spesifikasi Use-case History Data Transaksi (Guru)

Use Case	View Riwayat Transaksi
<i>Actor</i>	Guru
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses data transaksi siswa.
<i>Pre-conditions</i>	Setiap data transaksi tampil pada halaman list data transaksi.
<i>Post-conditions</i>	Klik button List Transaksi, tampil halaman data transaksi.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> <li>1. Mengakses aplikasi pendataan transaksi tabungan siswa-siswi.</li> <li>2. Melakukan konfirmasi transaksi.</li> </ol>
<i>Extensions / Alternate Scenario</i>	-
<i>Priority</i>	High
<i>Special Requirements</i>	-
<i>Open Questions</i>	-

**Tabel 1- 24.** Spesifikasi Use-case Cek Data Tabungan Siswa (Guru)

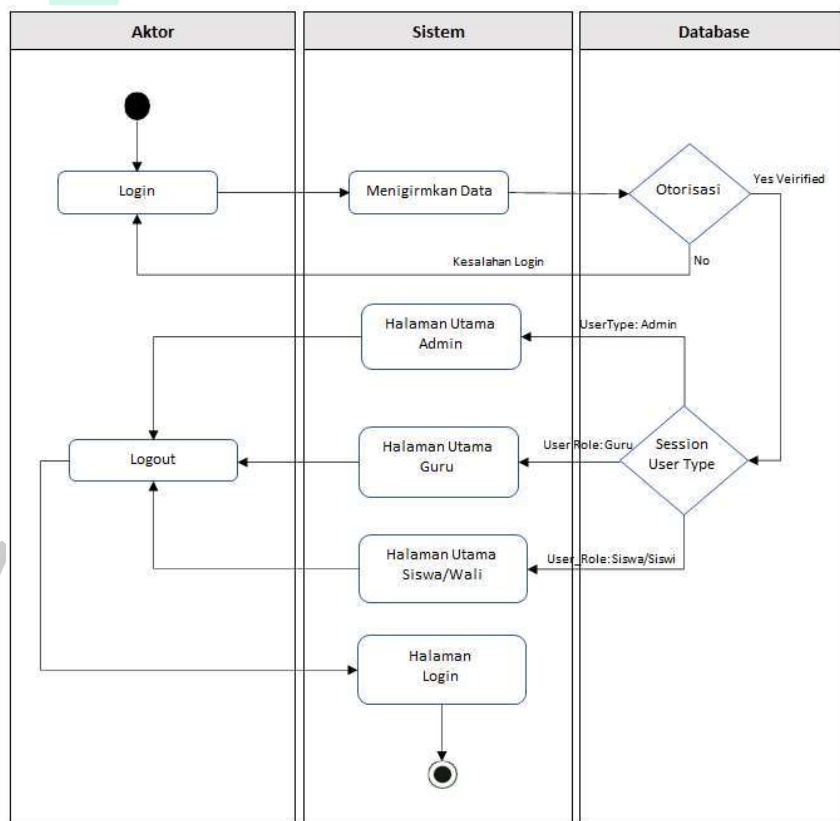
Use Case	View Data Tabungan
<i>Actor</i>	Guru
<i>Stakeholders and Interest</i>	-
<i>Trigger</i>	Mengakses data tabungan siswa.
<i>Pre-conditions</i>	Setiap data tabungan siswa tampil dalam data table.
<i>Post-conditions</i>	Klik button/tombol Data Tabungan, tampil halaman data tabungan.

<b>Main Success Scenario</b>	Mengakses aplikasi dan muncul data tabungan siswa-siswi dengan isi nama siswa, jumlah kredit, jumlah debit dan jumlah saldo.
<b>Extensions / Alternate Scenario</b>	-
<b>Priority</b>	High
<b>Special Requirements</b>	-
<b>Open Questions</b>	-

#### 4.3.2. Activity Diagram

*Activity diagram* menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang. Bagaimana masing-masing aktivitas yang terjadi pada aplikasi sistem informasi tabungan siswa dengan menggunakan sistem, sebagai berikut:

- Proses login ini dilakukan oleh semua aktor yaitu Petugas TU, Guru dan Wali Murid ketika mengakses halaman login, aktor harus memasukkan akun email dan password, maka sistem akan memvalidasi, jika email dan password yang dimasukan valid, maka sistem akan mengarahkan ke halaman utama sesuai dengan role masing-masing. sedangkan jika data yang masukan tidak valid, maka sistem akan menampilkan pesan kesalahan. Activity diagram login dapat dilihat dibawah ini.



**Gambar 1- 10.** Activity Diagram Login

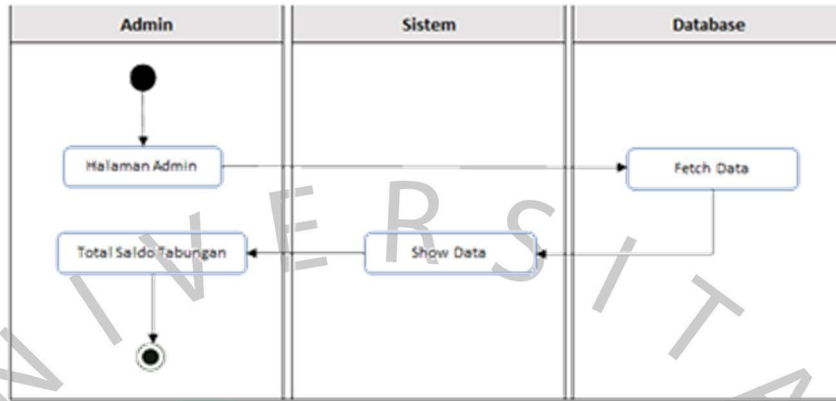
- Pada *activity diagram* ini adalah *activity diagram* yang dapat diakses oleh Admin. Beberapa tindakan yang dapat dilakukan admin adalah menampilkan data total saldo, mengelola data siswa, mengelola data account/pemilik tabungan dan mengelola data transaksi

Adapun Activity diagram pada akses admin adalah sebagai berikut:

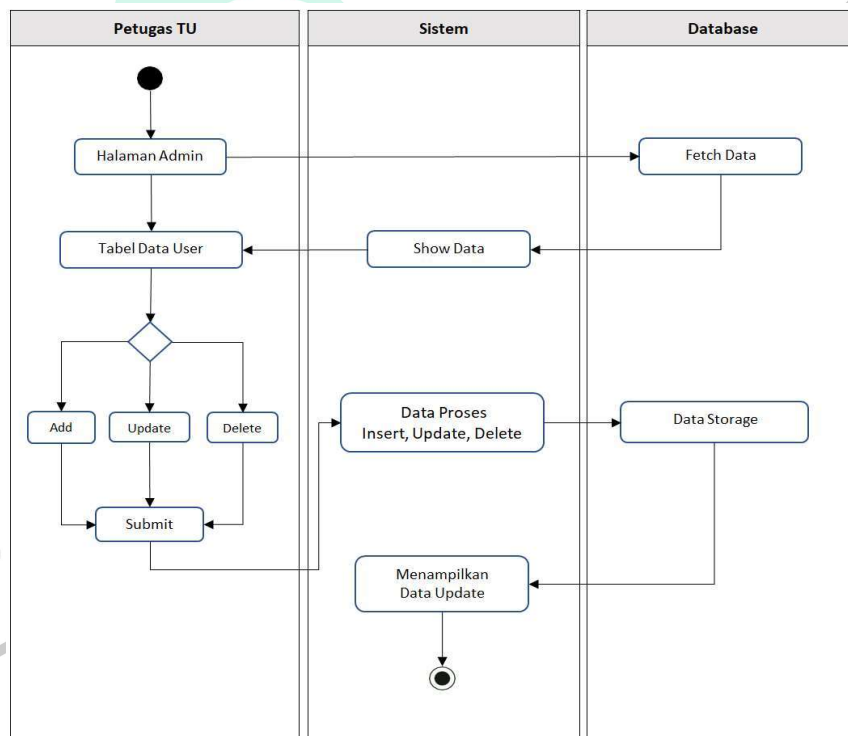
1. Menampilkan Total Saldo

2. Kelola Data User
3. Kelola Data *Account*
4. Kelola Data Transaksi

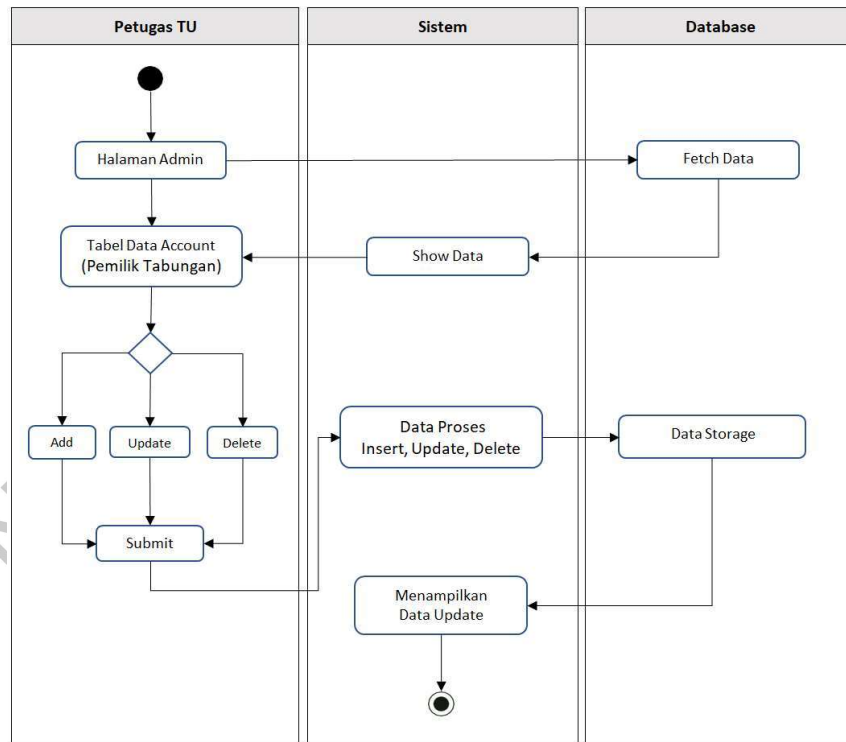
Activity diagram tersebut dapat dilihat dibawah ini.



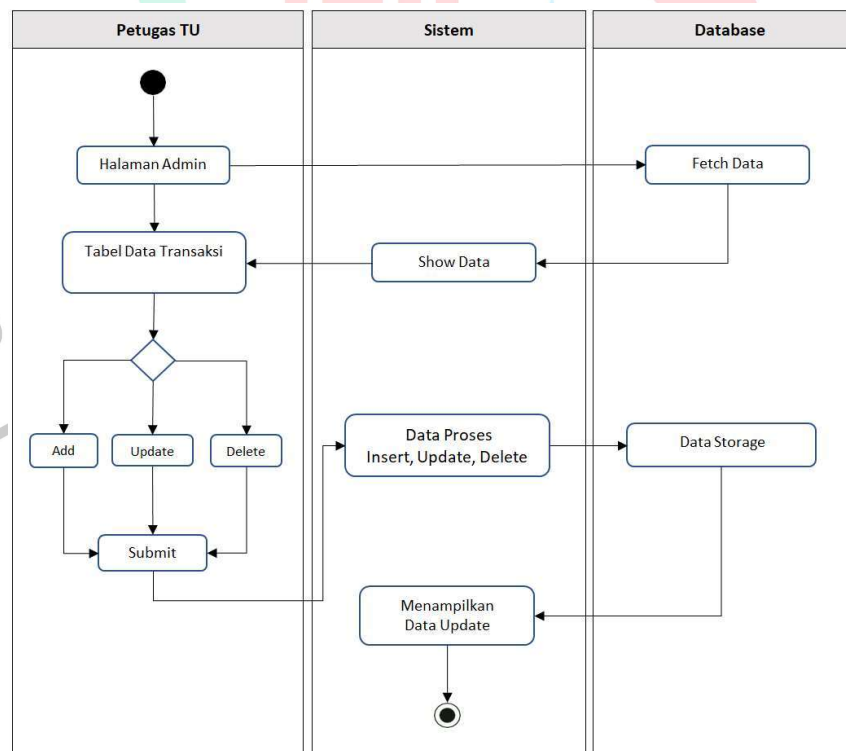
**Gambar 1- 11.** Activity Diagram Menampilkan Total Saldo



**Gambar 1- 12.** Activity Diagram Kelola Data User



Gambar 1- 13. Activity Diagram Kelola Data Account



Gambar 1- 14. Activity Diagram Kelola Data Transaksi

- Pada activity diagram berikutnya adalah activity diagram yang dapat diakses oleh Guru. Beberapa tindakan yang dapat dilakukan guru adalah menampilkan data total saldo, melihat

list transaksi dan melakukan konfirmasi akan transaksi serta melihat data tabungan.

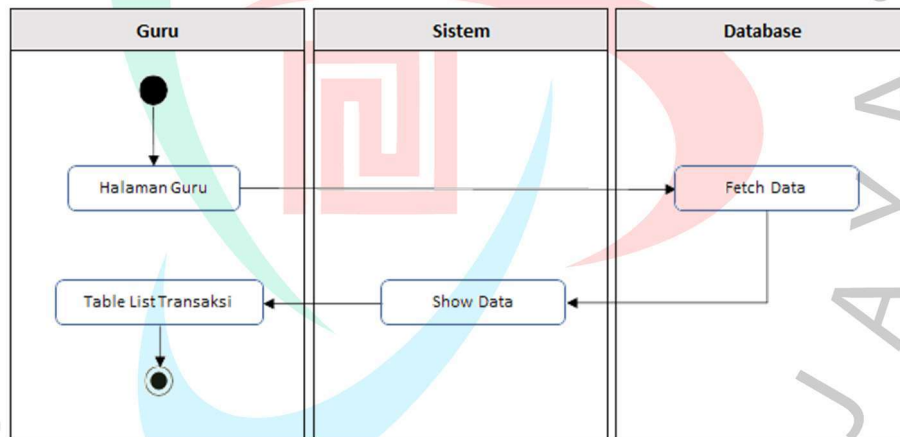
Adapun Activity diagram pada akses Guru adalah sebagai berikut:

1. Menampilkan Total Saldo
2. Melihat List Transaksi
3. Konfirmasi Isi Saldo
4. Konfirmasi Tarik Saldo
5. Konfirmasi Pembayaran
6. Melihat Data Tabungan Siswa

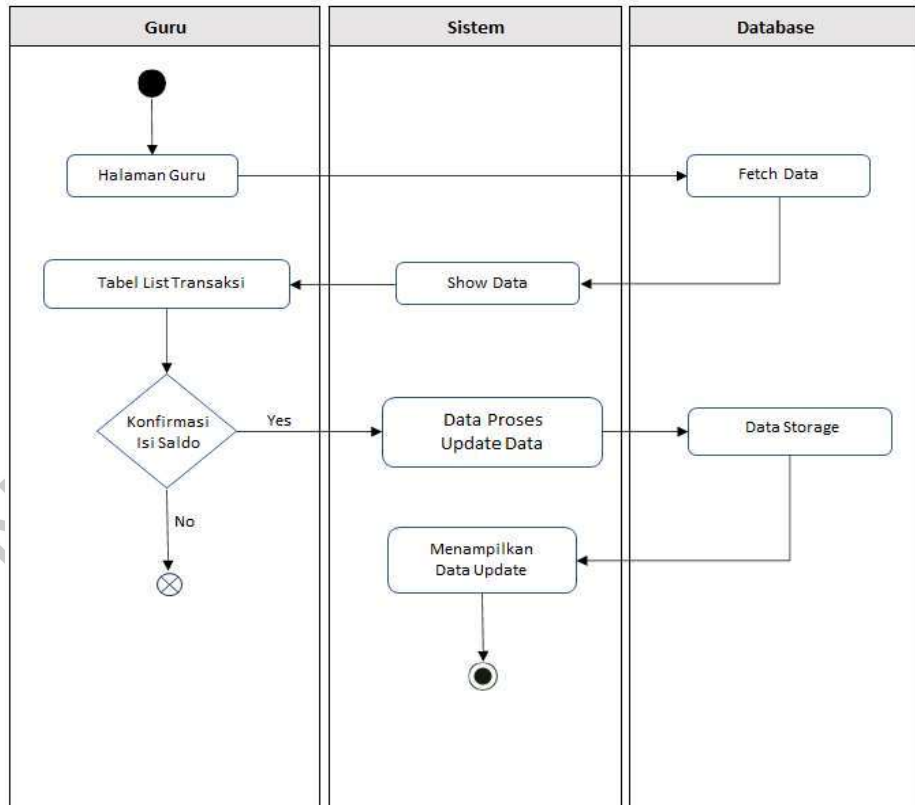
Activity diagram tersebut dapat dilihat dibawah ini.



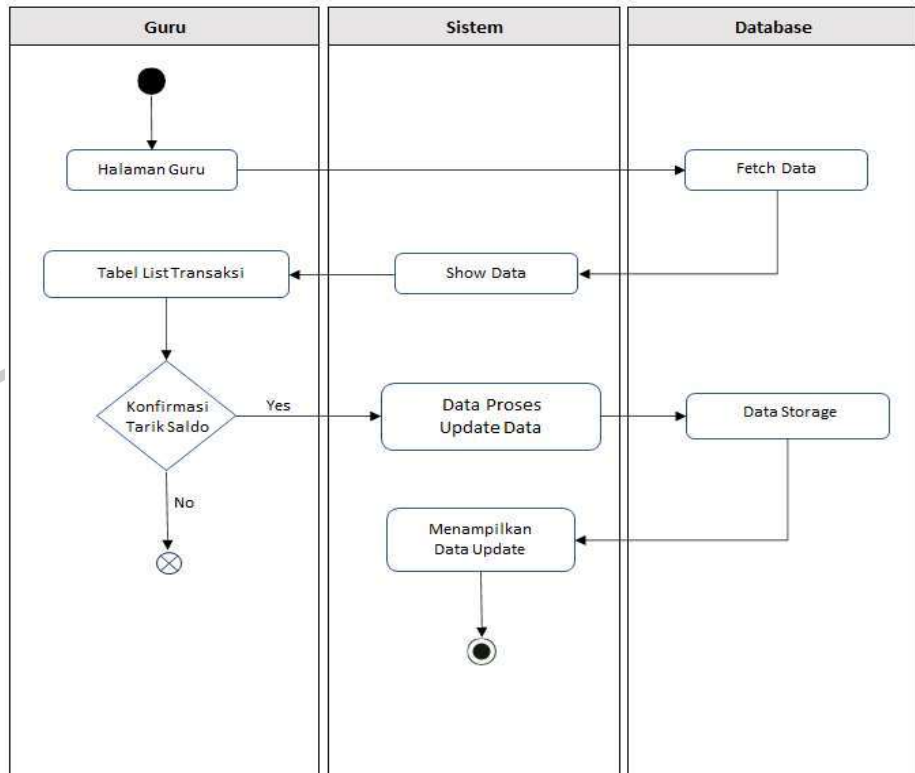
Gambar 1- 15. Activity Diagram Menampilkan Total Saldo



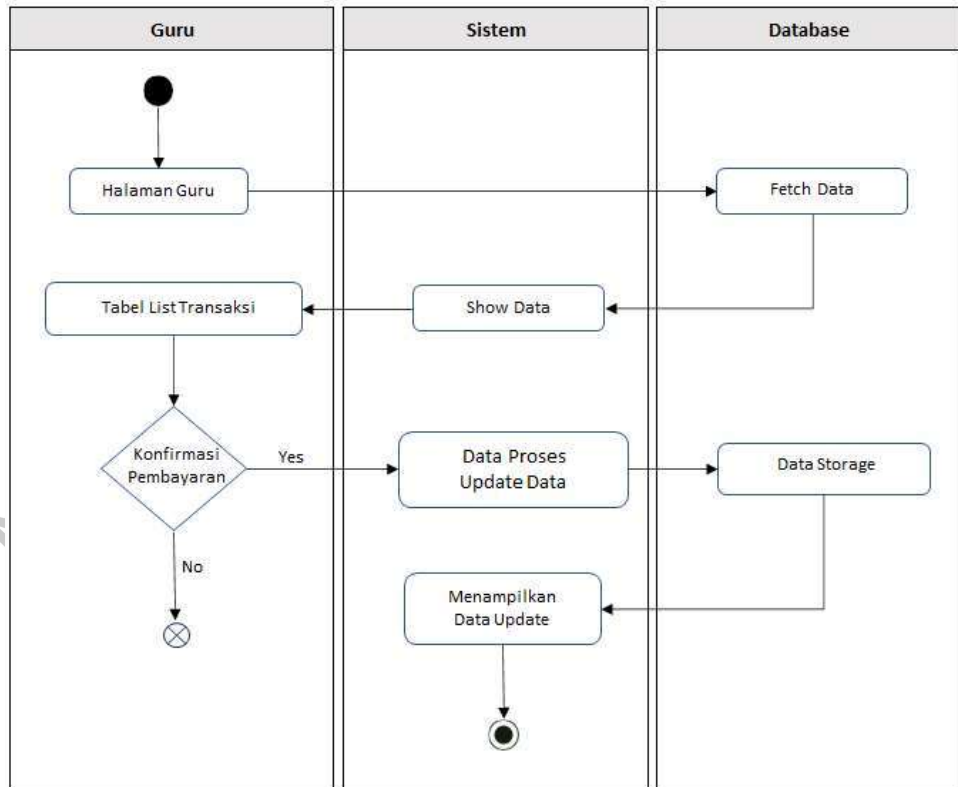
Gambar 1- 16. Activity Diagram Melihat List Transaksi



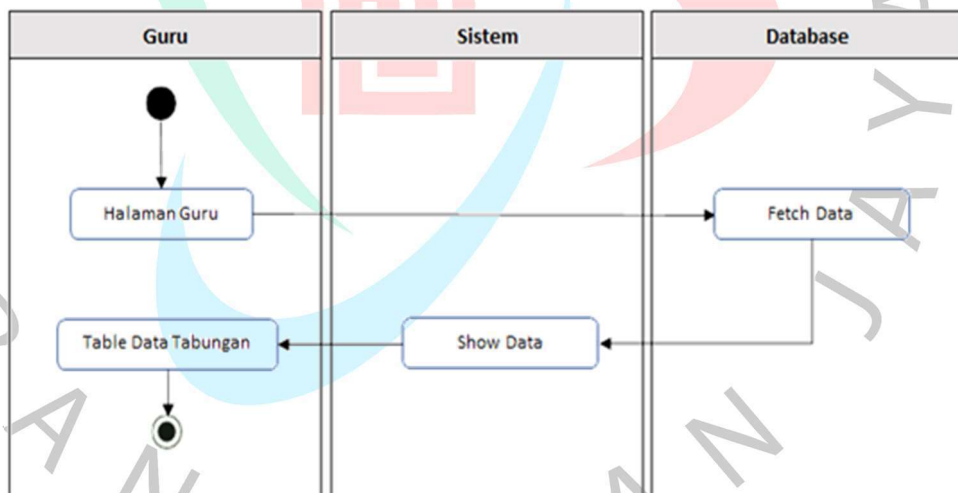
Gambar 1- 17. Activity Diagram Konfirmasi Isi Saldo



Gambar 1- 18. Activity Diagram Konfirmasi Tarik Saldo



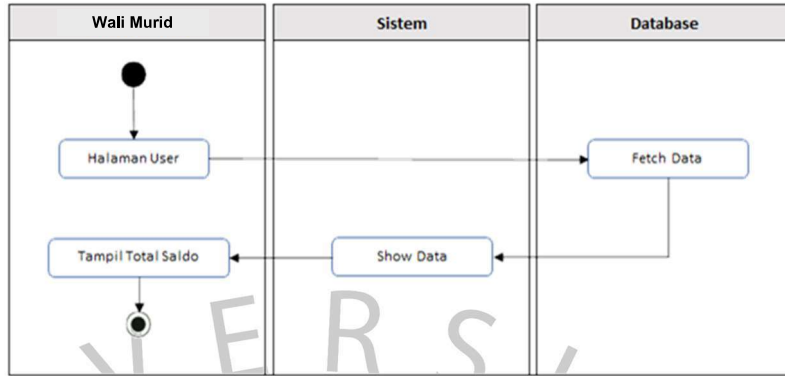
Gambar 1- 19. Activity Diagram Konfirmasi Pembayaran



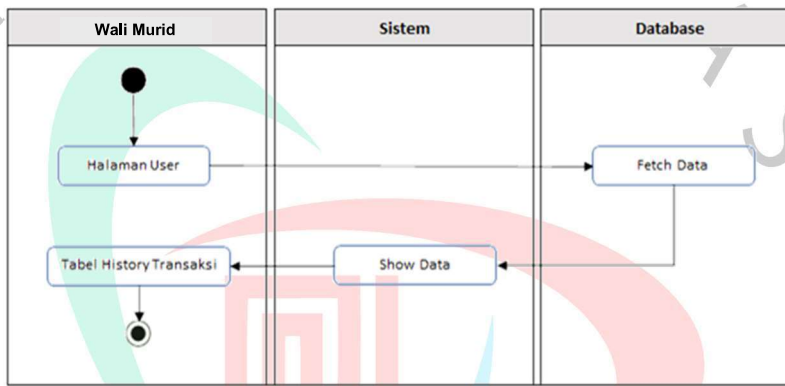
Gambar 1- 20. Activity Diagram Melihat Data Tabungan

- Pada *activity diagram* ini adalah *activity diagram* yang dapat diakses oleh Wali Murid. Beberapa tindakan yang dapat dilakukan oleh Wali Murid adalah melihat saldo, menampilkan *history* transaksi, input form isi saldo, input form tarik saldo, input form pembayaran. Adapun *Activity diagram* pada akses Wali Murid adalah sebagai berikut:
  1. Menampilkan Saldo
  2. Melihat Riwayat Transaksi
  3. Isi Saldo
  4. Tarik Saldo
  5. Pembayaran

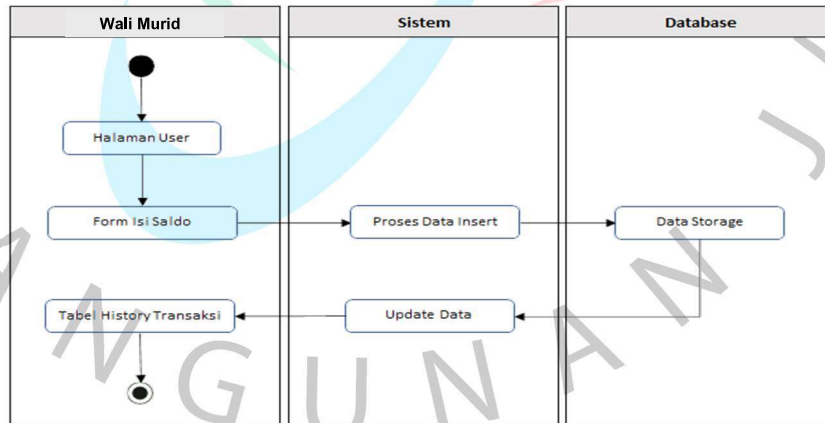
Activity diagram tersebut dapat dilihat pada gambar dibawah ini.



Gambar 1- 21. Activity Diagram Menampilkan Saldo

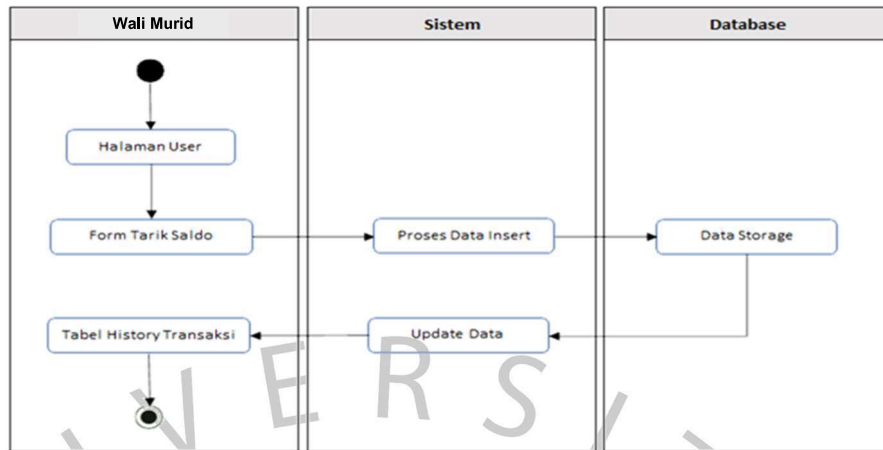


Gambar 1- 22. Activity Diagram Melihat History Transaksi

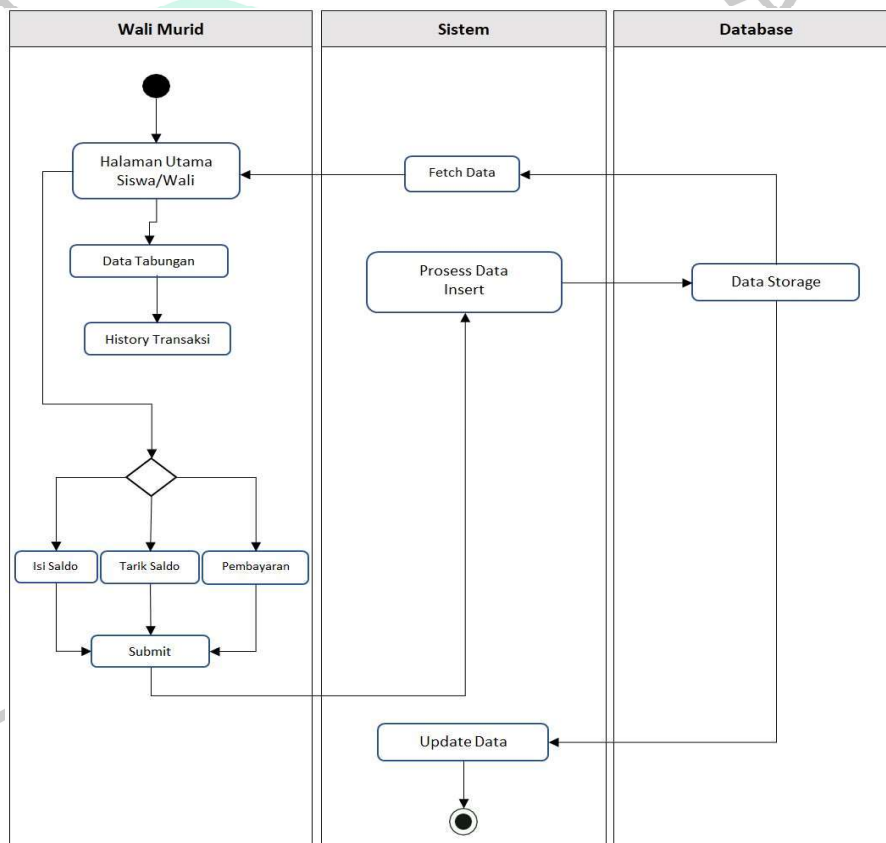


Gambar 1- 23. Activity Diagram Isi Saldo





Gambar 1- 24. Activity Diagram Tarik Saldo

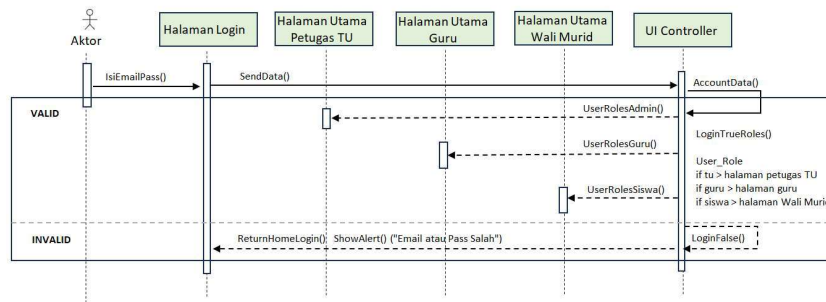


Gambar 1- 25. Activity Diagram Pembayaran

#### 4.3.3. Sequence Diagram

Sequence Diagram digunakan untuk menunjukkan aliran fungsional use case diagram, berikut adalah rancangan sequence diagram dari aplikasi tabungan siswa. Pada diagram sequence ini dapat diketahui urutan proses login terhadap setiap aktor.

Pada proses Login Aktor memiliki aturan atau roles ketika login yang akan diarahkan pada halaman sesuai dengan halaman akses. Sequence Login dapat dilihat pada gambar dibawah ini.



Gambar 1- 26. Sequence Diagram Login

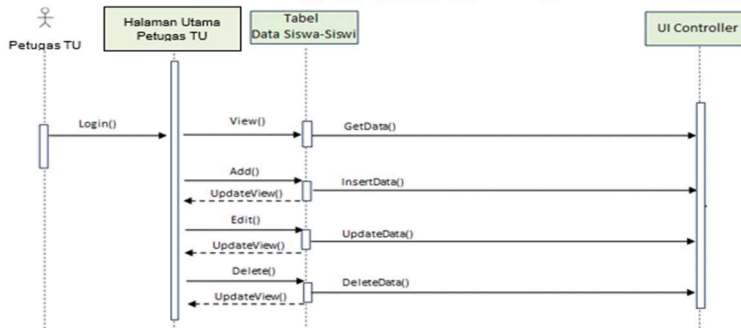
Pada *diagram sequence* berikutnya adalah diagram *sequence* terhadap aktor Petugas TU yang berisi akses role TU dengan aktivitas *Sequence* atau urutan proses pada petugas tu adalah sebagai berikut :

- MenampilkanTotal Saldo
- Kelola Data Siswa-Siswi
- Kelola Data *Account*
- Kelola Data Transaksi

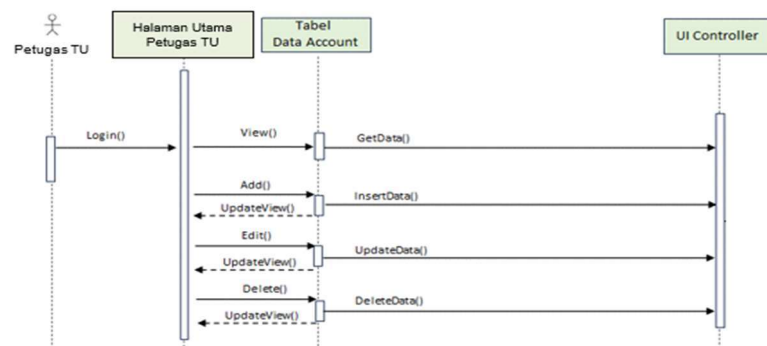
*Sequence diagram* tersebut dapat dilihat dibawah ini.



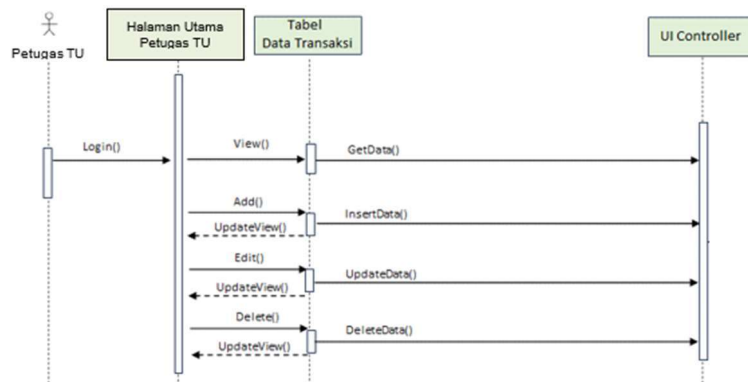
Gambar 1- 27. Sequence Diagram MenampilkanTotal Saldo



Gambar 1- 28. Sequence Diagram Kelola Data Siswa-Siswi



Gambar 1- 29. Sequence Diagram Kelola Data Account

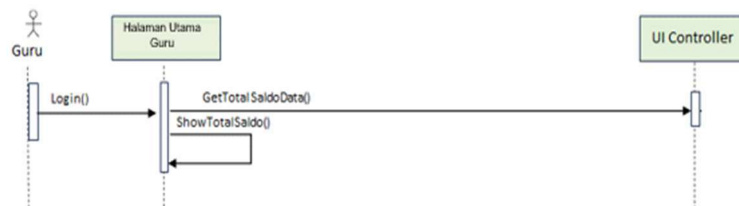


**Gambar 1- 30.** Sequence Diagram Kelola Data Transaksi

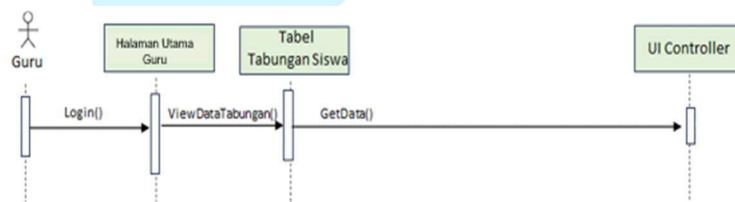
Pada diagram *sequence* dibawah ini merupakan diagram *sequence* terhadap aktor Guru yang berisi akses Guru dengan aktivitas *Sequence* atau urutan prosesnya. *Sequence* diagram yang ada pada guru adalah sebagai berikut:

- Menampilkan Total Saldo
- Melihat List Transaksi
- Konfirmasi Isi Saldo
- Konfirmasi Tarik Saldo
- Konfirmasi Pembayaran
- Melihat Data Tabungan Siswa

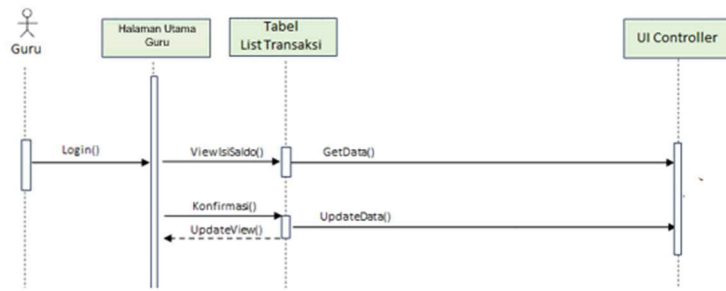
*Sequence diagram* tersebut dapat dilihat pada gambar dibawah ini.



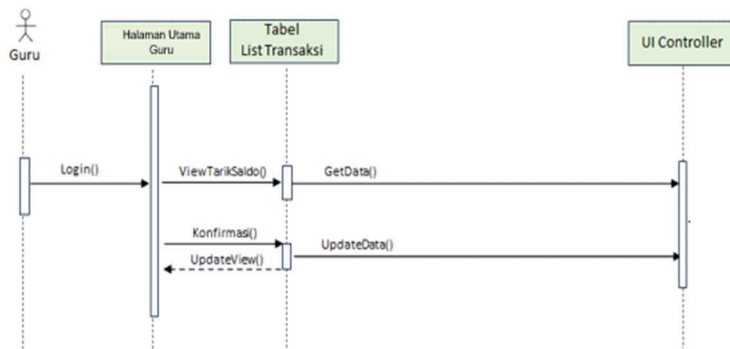
**Gambar 1- 31.** Sequence Diagram Menampilkan Total Saldo



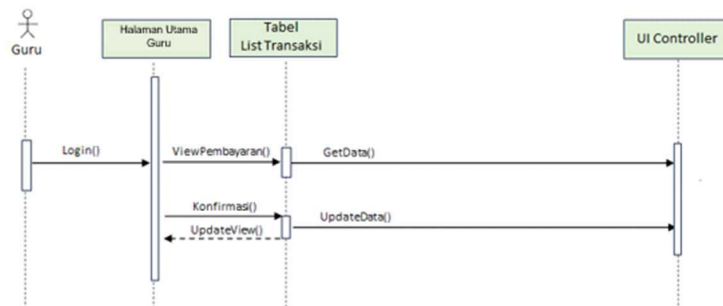
**Gambar 1- 32.** Sequence Diagram Melihat List Transaksi



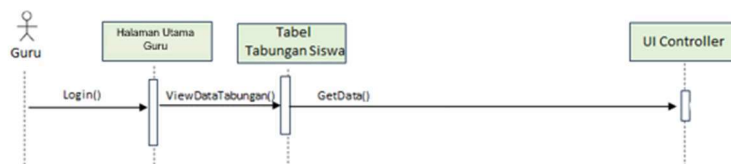
Gambar 1- 33. Sequence Diagram Konfirmasi Isi Saldo



Gambar 1- 34. Sequence Diagram Konfirmasi Tarik Saldo



Gambar 1- 35. Sequence Diagram Konfirmasi Pembayaran



Gambar 1- 36. Sequence Diagram Melihat Data Tabungan Siswa

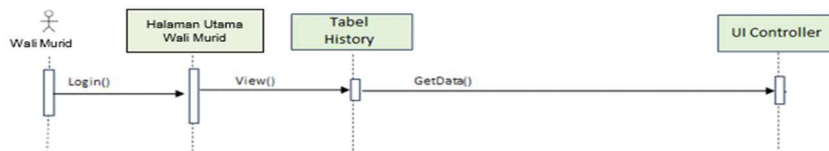
Pada diagram *sequence* ini adalah diagram sequence terhadap aktor Wali Murid yang berisi akses wali murid dengan aktivitas Sequence atau urutan prosesnya yang terdapat pada aktor user adalah sebagai berikut:

1. Menampilkan Saldo
2. Melihat History Transaksi
3. Isi Saldo
4. Tarik Saldo
5. Melakukan Pembayaran

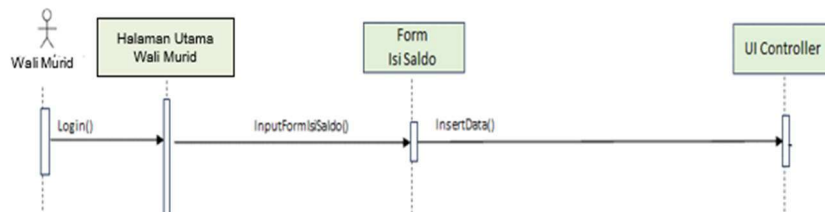
*Sequence diagram* tersebut dapat dilihat pada gambar dibawah ini.



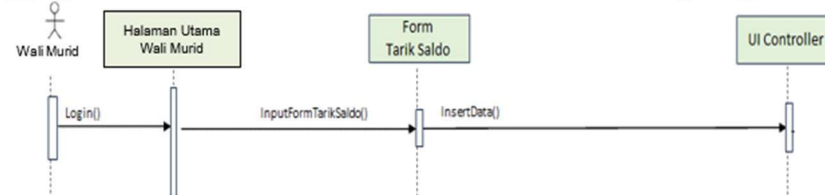
**Gambar 1- 37.** *Sequence Diagram* Menampilkan Saldo



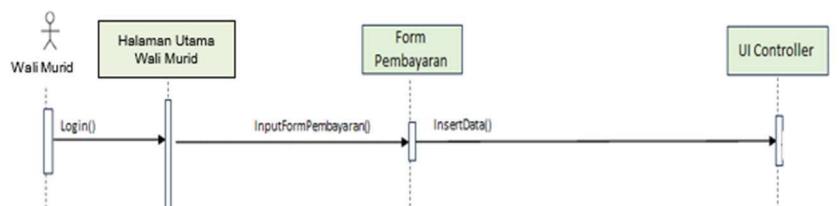
**Gambar 1- 38.** *Sequence Diagram* Melihat History Transaksi



**Gambar 1- 39.** *Sequence Diagram* Isi Saldo



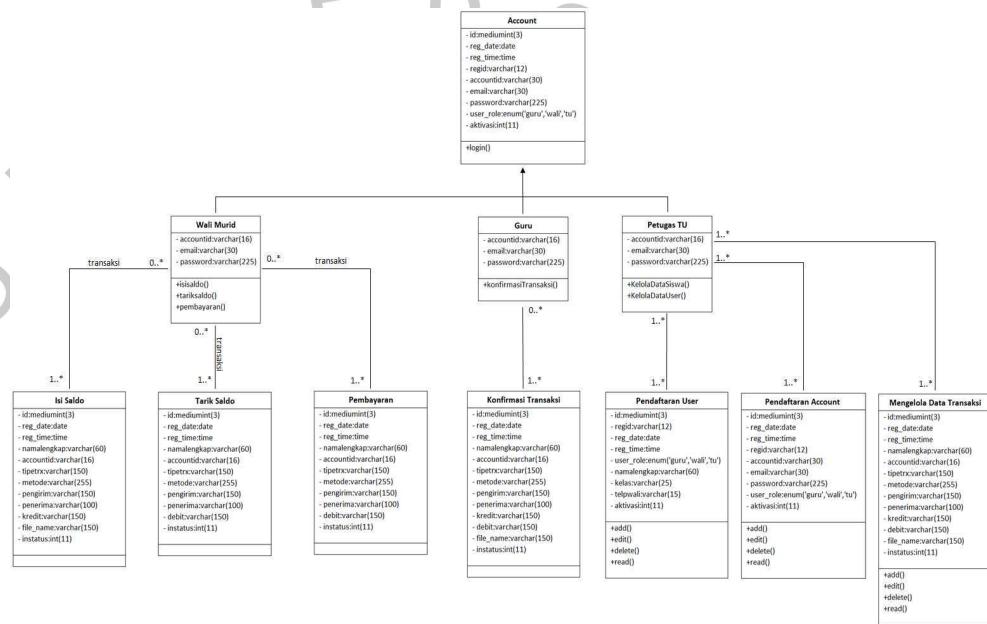
**Gambar 1- 40.** *Sequence Diagram* Tarik Saldo



Gambar 1- 41. Sequence Diagram Melakukan Pembayaran

4.3.4. Class Diagram

Class diagram memvisualisasikan atau menggambarkan keadaan atribut atau properti dalam suatu sistem, sekaligus menawarkan layanan dalam memanipulasi keadaan model tersebut. Berikut ini adalah class diagram dari aplikasi mobile tabungan siswa-siswi TK Ar-Rasyid yang dapat dilihat pada gambar dibawah ini.



Gambar 1- 42. Class Diagram Aplikasi Tabungan

4.3.5. Spesifikasi Basis Data

Desain persancangan basis data di jelaskan dalam tabel spesifikasi basis data berisi penjelasan tipe data atau informasi field terhadap table yang digunakan. Pada table Spesifikasi Basis Data akan menampilkan nama field, tipe dan ukuran field, field kunci (primary key) dan field unik (Unique Key). Berikut ini adalah tabel Spesifikasi Basis Data yang dibangun pada aplikasi:

Tabel 1- 25. Tabel Spesifikasi Basis Data Database User

Table Name: db\_user

No	Akronim	Nama Field	Type & Ukuran	Null	Keterangan
1	ID	id	mediumint(3)	No	Primary Key, Auto Increment
2	Nomor Registrasi	regid	varchar(100)	No	Uniqe Key
3	Posisi	posisi	enum(Guru, Wali, TU)	No	
4	Tanggal	reg_date	date	No	
5	Jam	reg_time	time	No	
6	Nama Panggilan	nickname	varchar(30)	No	

7	Nama Lengkap	namalengkap	varchar(60)	No	
8	Alamat	alamat	varchar(255)	No	
9	Nomor HP	notelp	varchar(255)	No	
10	Kelas	kelas	varchar(25)	No	
11	Tgl. Lahir	tlahir	date	No	
12	Jenis Kelamis	gender	varchar(171)	No	
13	Photo	photo	varchar(255)	No	
14	Telp Rumah	telpwali	varchar(15)	No	
15	Aktivasi	aktivasi	int(4)	No	

Tabel 1- 26. Tabel Spesifikasi Basis Data Database Pemilik Tabungan

Table Name: **db\_bukutabungan**

No		Nama Field	Type & Ukuran	Null	Keterangan
1	ID	id	mediumint(3)	No	Primary Key, Auto Increment
2	Tanggal	reg_date	date	No	
3	Jam	reg_time	time	No	
4	Nomor Registrasi	regid	varchar(100)	No	
5	Nomor Account	accountid	varchar(16)	No	Unique Key
6	Email	email	varchar(30)	No	
7	Password	password	varchar(255)	No	
8	User Level	user_role	enum(guru,wali,tu)	No	
9	Aktivasi	aktivasi	int(4)	No	

Tabel 1- 27. Tabel Spesifikasi Basis Data Database Transaksi

Table Name: **db\_transaksi**

No		Nama Field	Type & Ukuran	Null	Keterangan
1	ID	id	mediumint(6)	No	Primary Key, Auto Increment
2	Tanggal	reg_date	date	No	
3	Jam	reg_time	time	No	
4	Nama Lengkap	namalengkap	varchar(150)	No	
5	Nomor Account	accountid	varchar(32)	No	
6	Tipe Transaksi	tipetrx	varchar(150)	No	
7	Metode	metode	varchar(255)	No	
8	Pengirim	pengirim	varchar(150)	No	
9	Penerima	penerima	varchar(100)	No	
10	Kredit	kredit	varchar(150)	No	
11	Debit	debit	varchar(150)	No	
12	Keterangan	keterangan	varchar(150)	No	
13	File Name	file_name	varchar(150)	No	
14	Status	instatus	int(4)	No	

Dalam beberapa data yang akan ditampilkan diperlukan adanya integrasi data antar tabel data dan memerlukan perumusan perhitungan pada beberapa nilai khususnya dalam menghitung total transaksi dan menghitung saldo setiap pemilik tabungan. Untuk memenuhi data tersebut dibutuhkan teknik manipulasi data, penulis menggunakan bahasa substandard SQL (*Structure Query Language*) dengan cara membangun virtual table pada database serta penggabungan data dan manipulasi data. Pada perancangan ERD membuat *virtual table* dapat menggunakan perintah *Data Definition Language* (DDL) dengan perintah CREATE VIEW yang akan membentuk tabel virtual dan penggabungan data menggunakan *Data Query Language* (DQL) dengan perintah JOIN serta SQL AGGREGATE FUNCTION yaitu menghitung jumlah (SUM). Berikut ini adalah hasil data table virtual:

**Tabel 1- 28.** Tabel *Virtual Database* Total Saldo

Structure for view **view\_ttltransaksi**

No	Nama Field	Type & Ukuran	Keterangan
1	totalkredit	varchar(4)	
2	totaldebit	varchar(4)	
3	totalsaldo	varchar(4)	

**Tabel 1- 29.** Tabel *Virtual Database* Saldo Pemilik Tabungan

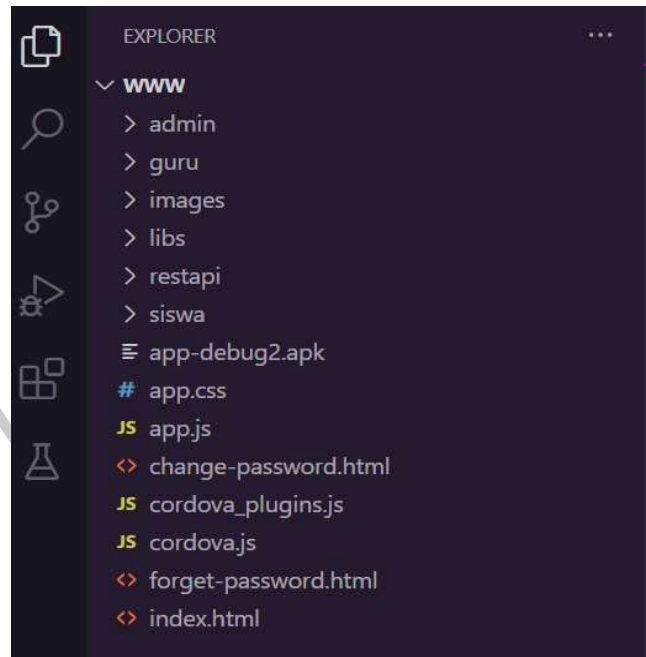
Structure for view **view\_saldouser**

No	Nama Field	Type & Ukuran	Keterangan
1	id	int(11)	
2	reg_date	date	
3	reg_time	time	
4	accountid	varchar(30)	
5	namalengkap	varchar(60)	
6	email	varchar(30)	
7	sumkredit	varchar(418)	
8	sumdebit	varchar(418)	
9	saldo	varchar(418)	
10	user_role	enum(admin,guru,siswa)	
11	instatus	int(4)	

#### 4.4. Membangun Prototype Awal

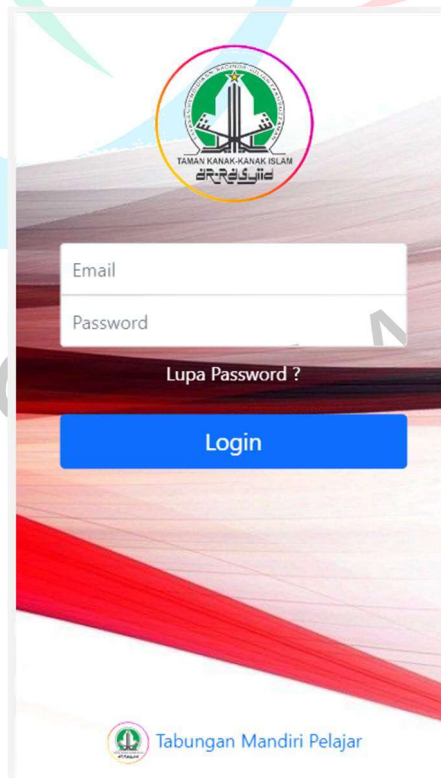
Prototipe yang dibangun pada tahap pertama akan dibangun dengan pemograman berbasis web dengan batasan tampilan *responsive* terhadap *mobile device* serta dengan penggunaan bahasa *JavaScript* dan penggunaan *server localhost*. Adapun *structure* pemograman adalah sebagai berikut:





**Gambar 1- 43.** *Structure* Aplikasi

Prototipe dikembangkan dalam tampilan *User Interface* (UI) pada pemrograman berbasis web dengan tampilan terhadap tiga aktor yaitu Admin, Guru dan Siswa/Siswi. Hasil Prototipe pada tampilan *User Interface* adalah sebagai berikut:

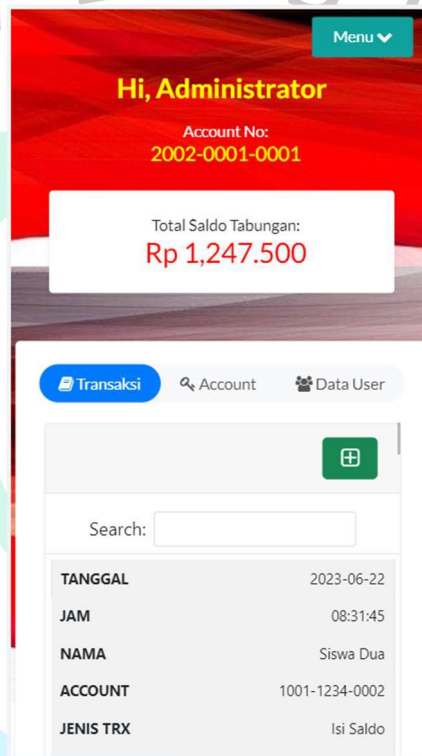


**Gambar 1- 44.** UI Login

**A. UI User Petugas TU**

**1. Tampilan Utama Petugas TU**

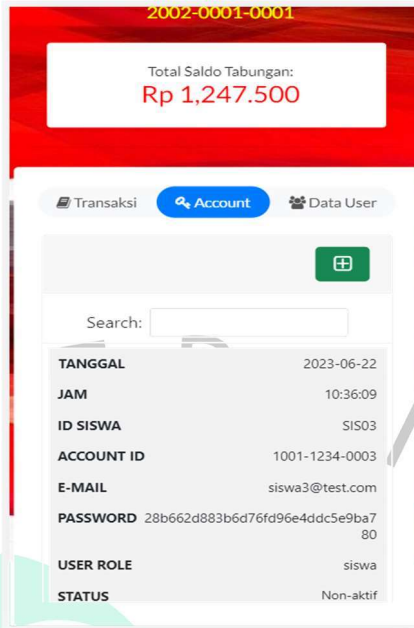
Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan total tabungan siswa, dimana antarmuka tersebut memaparkan tentang menu apa saja yang bisa di dapatkan oleh admin, menu-menu yang dapat mengelola data transaksi, data *account* dan data siswa/siswi, dapat dilihat pada gambar dibawah ini.



**Gambar 1- 45.** UI Tampilan Utama Admin

**2. Tampilan Kelola Data *Account***

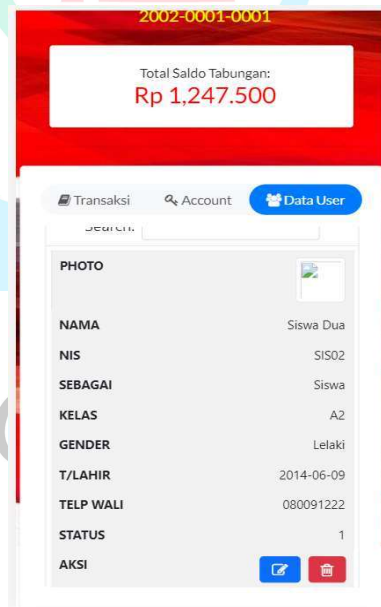
Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan tabungan siswa, dimana antarmuka tersebut memaparkan tentang pengelolaan pengguna yang dapat dilakukan tambah data, edit data yang berisikan data pengguna, dapat dilihat pada gambar dibawah ini.



**Gambar 1- 46.** UI Tampilan Kelola Data *Account*

3 Tampilan Kelola Data Siswa

Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan tabungan siswa, dimana antarmuka tersebut memaparkan tentang pengelolaan siswa yang dapat dilakukan tambah data, edit data yang berisikan data siswa, dapat dilihat pada gambar dibawah ini.

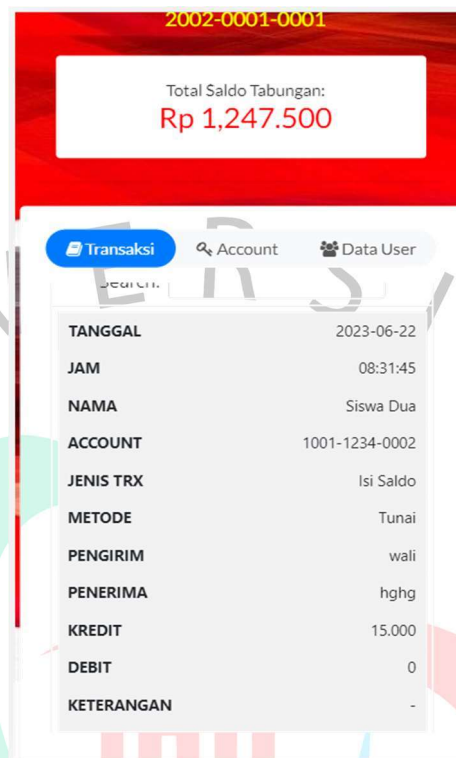


**Gambar 1- 47.** UI Tampilan Kelola Siswa

4 Tampilan Kelola Data Transaksi

Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan transaksi

siswa, dimana antarmuka tersebut memaparkan tentang pengelolaan tabungan yang dapat dilakukan tambah data, edit data yang berisikan data tabungan, dapat dilihat pada gambar dibawah ini.

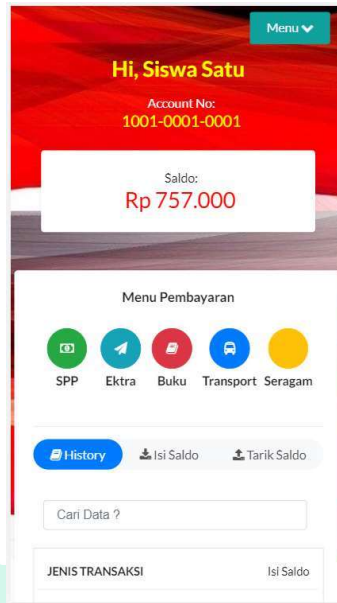


Gambar 1- 48. UI Tampilan Kelola Data Transaksi

## B. UI User Wali Murid

### 1. Tampilan Utama Wali Murid

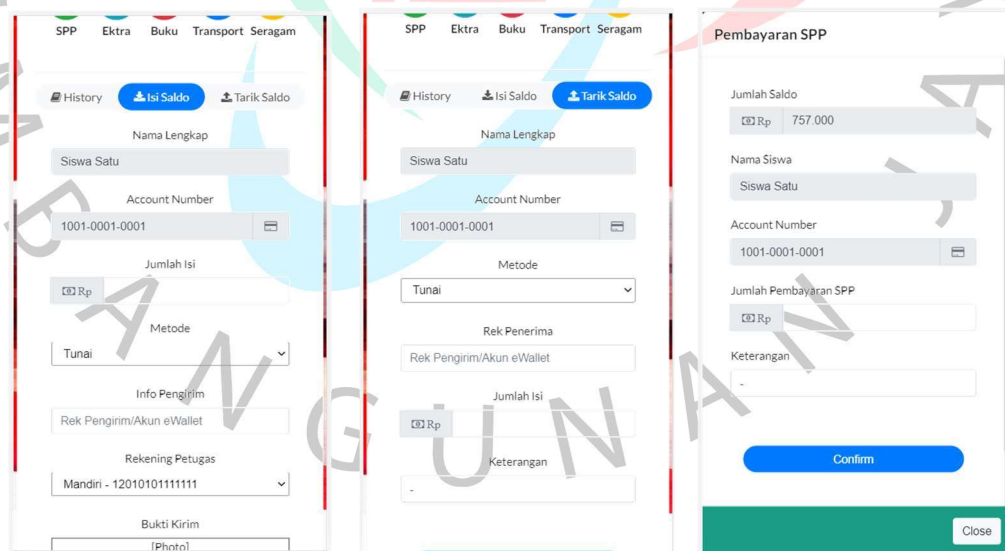
Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan tabungan siswa, dimana antarmuka tersebut memaparkan tentang *menu* apa saja yang bisa di dapatkan oleh siswa, terdapat menu data *history*, *form* isi saldo, *form* tarik saldo serta tombol menu pembayaran, dapat dilihat pada gambar dibawah ini.



**Gambar 1- 49.** UI Tampilan Utama *User* atau Siswa/Siswi

## 2. Tampilan *Form* Transaksi

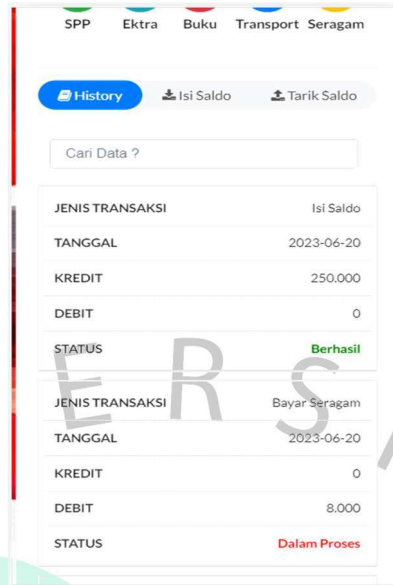
Halaman transaksi tabungan berisi informasi mengenai *form* isi saldo, tarik saldo dan pembayaran. Pada *menu* ini pengguna dapat memilih jenis transaksi yang akan dilakukan, sistem akan menampilkan *form* tergantung dengan jenis transaksi yang dilakukan yaitu transaksi isi saldo, transaksi tarik saldo dan transaksi pembayaran, tampilan halaman dapat dilihat pada gambar dibawah ini.



**Gambar 1- 50.** UI Tampilan Transaksi

## 3. Tampilan *History* Transaksi

Pada gambar berikut merupakan rancangan antarmuka akan tampilan tabel sejarah transaksi.



**Gambar 1- 51.** UI Tampilan *History* Transaksi

### C. UI User Guru

#### 1. Tampilan Utama Guru

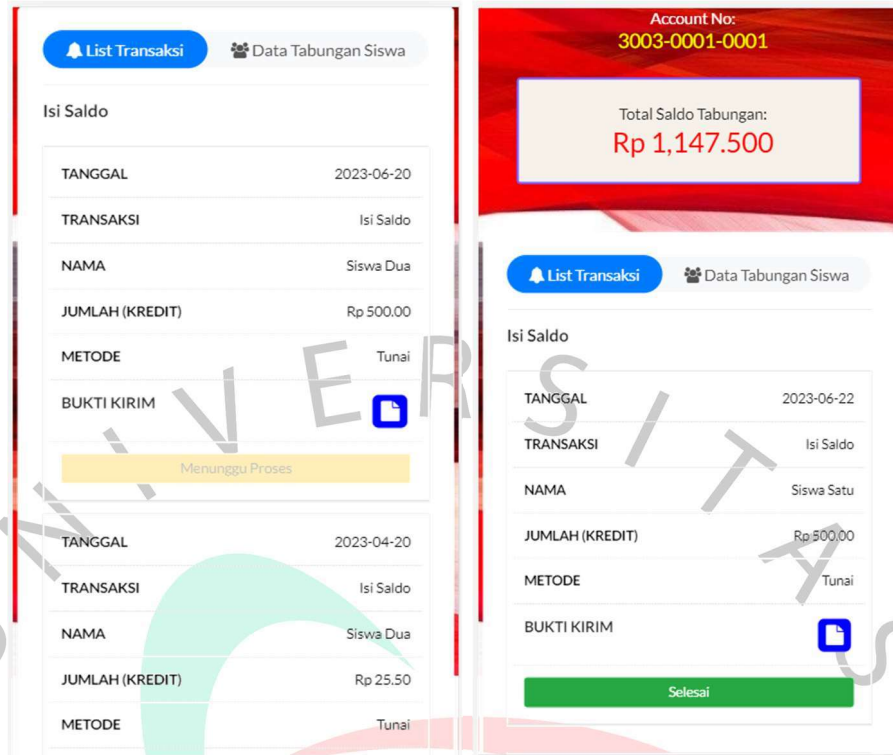
Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada pendataan total tabungan siswa dan terdapat menu list transaksi dan data tabungan siswa, dapat dilihat pada gambar dibawah ini.



**Gambar 1- 52.** UI Tampilan Utama Guru

#### 2. Tampilan *List* Transaksi

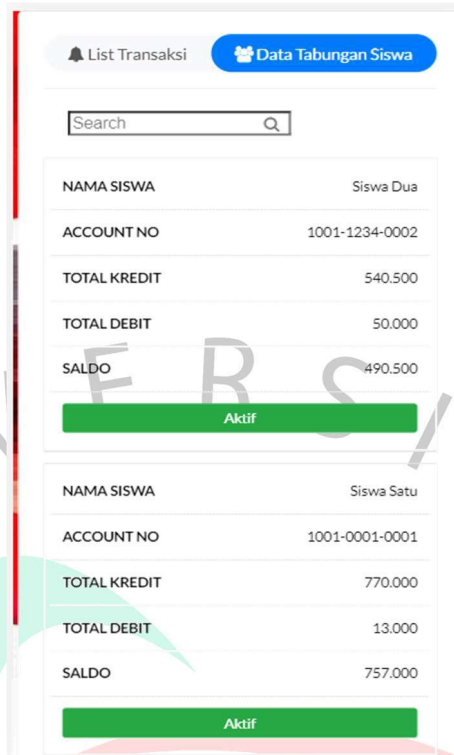
Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada *list* data transaksi siswa/siswi sebagai notifikasi yang harus di respon oleh guru yang mana terdapat tombol konfirmasi, jika terdapat permintaan transaksi dari siswa maka terdapat tombol “Menunggu Proses” dan jika sudah dikonfirmasi tombol akan menjadi “Selesai”. Tampilan dapat dilihat pada gambar dibawah ini.



**Gambar 1- 53.** UI Tampilan *List Transaksi*

### 3. Tampilan Data Tabungan Siswa/Siswi

Pada gambar di bawah ini merupakan sebuah gambar rancangan antarmuka pada data tabungan seluruh data siswa/siswi yang berisi status total saldo, total kredit, total debit dan status *account*. Tampilan dapat dilihat pada gambar dibawah ini.



**Gambar 1- 54.** UI Data Tabungan Siswa/Siswi

#### 4.5. Evaluasi *Prototype Awal*

Evaluasi Prototipe dilakukan dengan menggunakan data-data simulasi dan melibatkan wali murid, guru dan petugas tu Berikut ini adalah hasil evaluasi prototipe awal:

**Tabel 1- 30.** Hasil Evaluasi Prototype Awal

No	Evaluasi	Deskripsi Penilaian	Review
1	Tampilan	Tampilan baik dengan menggunakan <i>browser</i> , akan tetapi tampilan dapat berubah ketika pengguna <i>hanphone</i> merubah seting ukuran Font maka tampilan akan ikut berubah.	Akan dibuatkan CSS container dan <i>body zoom</i> untuk mengurangi perubahan besar pada tampilan.
2	Wali Murid dapat melihat Riwayat	Tabel dapat menampilkan data <i>history</i> transaksi.	
3	Wali Murid melakukan isi form isi saldo	<i>Form</i> semua dapat terisi dan terkirim dan dapat mengupload foto ketika melakukan isi saldo <i>transfer</i> dengan upload foto bukti <i>transfer</i> .	
4	Wali Murid melakukan isi Form tarik saldo	<i>Form</i> dapat terisi dan mengirimkan informasi tarik saldo.	
5	Wali Murid melakukan pembayaran	Wali Murid dapat mengakses halaman pembayaran dan dapat mengisi <i>form</i> pembayaran. Saldo akan berkurang ketika melakukan proses pembayaran.	



6	Guru melihat total saldo	Guru dapat melihat jumlah total saldo.	
7	Guru melakukan konfirmasi transaksi	Terdapat status transaksi yang dapat diproses oleh guru, tetapi baiknya ketika status masih proses tombol dibuat berkedip.	Akan ditambahkan animasi CSS berkedip/blink pada tombol ketika menunggu proses.
8	Guru melihat data status tabungan siswa	Data tabungan siswa dapat diakses dan data memunculkan data kredit, debit dan total saldo siswa.	
9	Petugas TU mengelola data transaksi	Petugas TU dapat melakukan <i>add</i> , <i>update</i> dan <i>delete</i> pada data transaksi.	
10	Petugas TU mengelola data <i>account</i>	Petugas TU dapat melakukan <i>add</i> , <i>update</i> dan <i>delete</i> pada data <i>account</i> .	
11	Petugas TU mengelola data <i>user</i>	Petugas TU dapat melakukan <i>add</i> , <i>update</i> dan <i>delete</i> pada data <i>user</i>	

Terdapat dua (2) perbaikan dari hasil evaluasi yaitu:

1. Tampilan *responsive* mengikuti seting *handphone*, dari hal tersebut maka akan menggunakan *class container* untuk mengurangi perubahan yang signifikan. Perbaikannya yaitu:

```
<div class="container header">
  content
</div>
```

2. Tombol konfirmasi guru ketika “Menunggu Proses” dibuat berkedip sebagai indikator penanda. Perbaikannya akan membuat animasi *class blink* pada CSS. *Class blink* adalah sebagai berikut:

```
<style>
.blink {
  animation: blinker 0.9s linear infinite;
}
@keyframes blinker {
  50% {
    opacity: 0;
  }
}
.blink-one {
  animation: blinker-one 1s linear infinite;
}
@keyframes blinker-one {
  0% {
    opacity: 0;
  }
}
.blink-two {
  animation: blinker-two 1.4s linear infinite;
}
@keyframes blinker-two {
  100% {
    opacity: 0;
  }
}
@-webkit-keyframes blink {
  0% {
    opacity: 1;
  }
}
```

```

50% {
  opacity:0;
}
100% {
  opacity:1;
}
}
@-moz-keyframes blink {
0% {
  opacity:1;
}
50% {
  opacity:0;
}
100% {
  opacity:1;
}
}
}
.blinking {
color: red;
background-color: red;
-webkit-transition: all 1s ease-in-out;
-moz-transition: all 1s ease-in-out;
-o-transition: all 1s ease-in-out;
-ms-transition: all 1s ease-in-out;
transition: all 1s ease-in-out;
-webkit-animation-direction: normal;
-webkit-animation-duration: 2s;
-webkit-animation-iteration-count: infinite;
-webkit-animation-name: blink;
-webkit-animation-timing-function: ease-in-out;
-moz-animation-direction: normal;
-moz-animation-duration: 2s;
-moz-animation-iteration-count: infinite;
-moz-animation-name: blink;
-moz-animation-timing-function: ease-in-out;
}
</style>

```

#### 4.6. Pemograman Sistem

Aplikasi yang dibangun adalah aplikasi mobile yang akan menggunakan *native engine* dengan teknik *Cross-platform Native Mobile Applications*. Dengan teknik *Cross-platform Native Mobile Applications* yang akan *rendering script HTML5, JavaScript* dan *Rest-API* dapat di ubah menjadi *Application Package Kit (APK)* Android yang dapat instalasi atau dipasangkan terhadap *mobile device* berbasis OS *Android*.

Pemograman sistem akan menggunakan *SDK Tools* dengan bantuan *Framework Apache Cordova* yang akan mengubah *script* Prototipe awal menjadi Prototipe yang diharapkan oleh pengguna. Instalasi yang dibutuhkan adalah sebagai berikut:

- Install Node.js
  - Install JDK (*Java Development Kit*)
  - Install Android Studio dan install Android SDK (*Software Development Kit*)
  - Install *System Kernel Linux GIT (Group Inclusive Tour)*
  - Install *Build Android Tools Graddle*
  - Lakukan *Setup Environment Variable* pada *Microsoft Windows*
- ✓ JAVA\_HOME : C:\Program File\Java\jdk1.8.0.241

- ✓ ANDROID\_HOME: C:\User\{user}\AppData\Local\Android\Sdk
- ✓ Path : C:\Users\{user}\AppData\Roaming\npm

Pemrograman rendering menjadi APK adalah sebagai berikut:

```

MINGW64:/c/xampp/htdocs/tabsirra/Ta-Wa
$ cordova plugin add cordova-plugin-badge
Adding cordova-plugin-badge to package.json

Mohammad Shohib@DESKTOP-04E17GT MINGW64 /c/xam
$ cordova platform add android
Using cordova-fetch for cordova-android
Adding android project...
Creating Cordova project for the Android platf
  Path: platforms\android
  Package: com.tabungansiswa.myapps
  Name: Ta-Wa
  Activity: MainActivity
  Android Target SDK: android-33
  Android Compile SDK: 33
Subproject Path: CordovaLib
Subproject Path: app
Android project created with cordova-android@12.
Installing "cordova-plugin-badge" for android
Subproject Path: CordovaLib
Subproject Path: app
Installing "cordova-plugin-device" for android
Installing "cordova-plugin-local-notification" f
android-sdk version check failed (C:\xampp\htdoc
ersion), continuing anyways.

are <{<codename>,<{<layoutlib>,<{<api-level>
> Task :CordovaLib:compileDebugJavaWithJavac
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

> Task :app:compileDebugJavaWithJavac
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.
You can use '--warning-mode all' to show the individual deprecation warnings and determine if the
See https://docs.gradle.org/7.1.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 10m 46s
48 actionable tasks: 48 executed
Built the following apk(s):
  E:\TabunganSiswa\platforms\android\app\build\outputs\apk\debug\app-debug.apk

Mohammad Shohib@DESKTOP-04E17GT MINGW64 /e/TabunganSiswa
$

```

Gambar 1- 55. Implementasi Android Package Kit

Database dan RestAPI di setup pada *server* dan setiap koneksi RestAPI dikonfigurasi dengan menambahkan *header* CORS (*Cross-Origin Resource Sharing*). Setup penambahan header CORS adalah sebagai berikut:

- `header("Content-type:application/json");`
- `header("Access-Control-Allow-Origin: *");`
- `header("Access-Control-Allow-Methods: PUT");`
- `header("Access-Control-Allow-Methods: GET");`
- `header("Content-Type: application/json; charset=UTF-8");`
- `header("Access-Control-Allow-Methods: OPTIONS,GET,POST,PUT,DELETE");`

- `header("Access-Control-Max-Age: 3600");`
- `header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");`

#### 4.6.1. Spesifikasi Perangkat Lunak

Dalam membangun aplikasi *mobile*, spesifikasi perangkat lunak yang dibutuhkan adalah sebagai berikut:

- Sistem Operasi Microsoft Windows 10 – 64BIT.
- Notepad++ GPL Version 4.4.
- Xampp Version 7.4.29-1-VC15 – 64BIT.
- phyMyAdmin 5.2.0.
- Android Studio Android Studio Dolphin | 2021.3.1.
- Cordova version 9.2.0
- GIT 2.39.0 -64BIT.
- Graddle 8.1.
- Java Jdk1.8.0\_341 dan Jre1.8.0\_341.
- Server Host DomaiNesia.
- Android Version Funtouch OS 10.5.
- Bluestack Version 5.

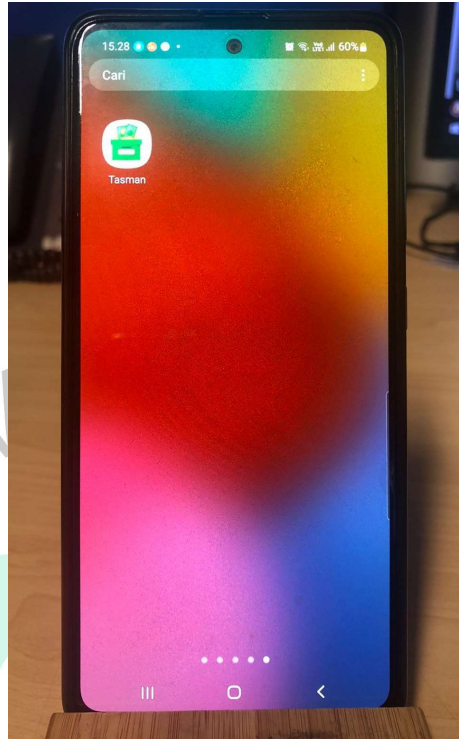
#### 4.6.2. Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

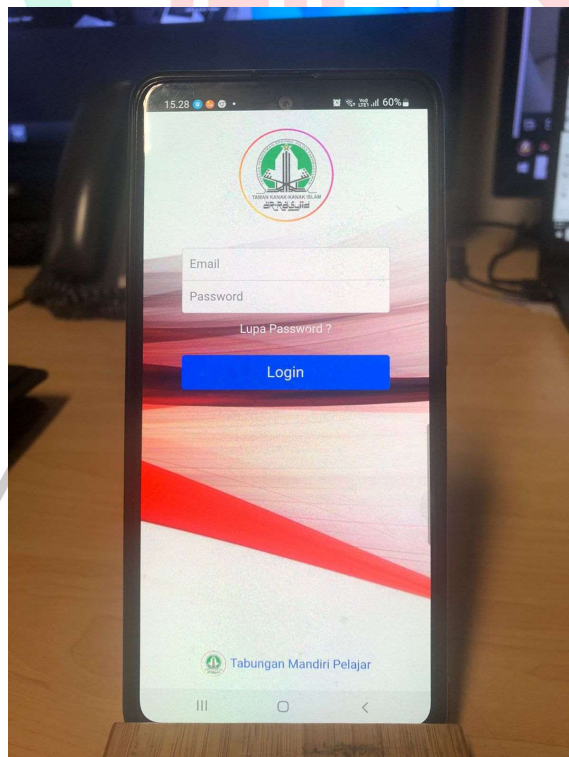
- Laptop HP Intel i5 2.80Ghz, RAM 8GB, SSD 512GB.
- HP VIVO 2.3Ghz Octa-core, RAM 4GB/64GB.

#### 4.6.3. Implementasi Prototype Akhir

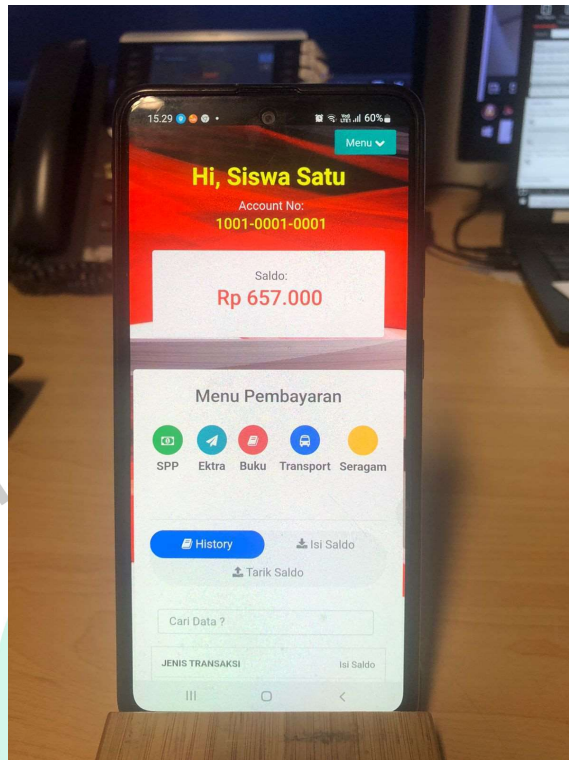
Prototype akhir adalah aplikasi berbasis *mobile* yang dapat dipasangkan atau terinstal pada telepon pintar (*smartphone*) berbasis operasi sistem Android. Berikut ini adalah hasil tampilan dalam penggunaan *Aplikasi Tabungan Siswa-Siswi TK Ar-Rasyid Berbasis Mobile* yang di pasangkan atau diinstal menggunakan telepon pintar dengan operasi sistem Android dan *Emulator Android* dengan *Bluestack* :



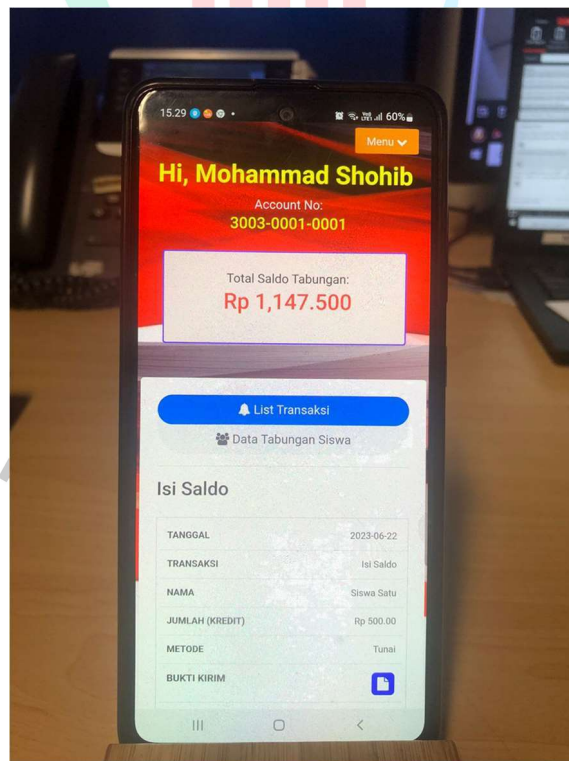
**Gambar 1- 56.** Implementasi Aplikasi Pada *Smartphone* Android



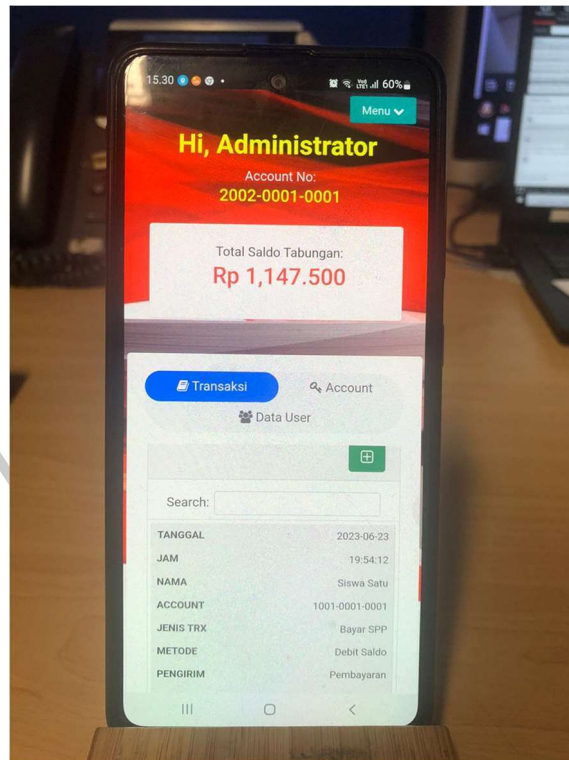
**Gambar 1- 57.** Implementasi Tampilan *Login* pada *Smartphone*



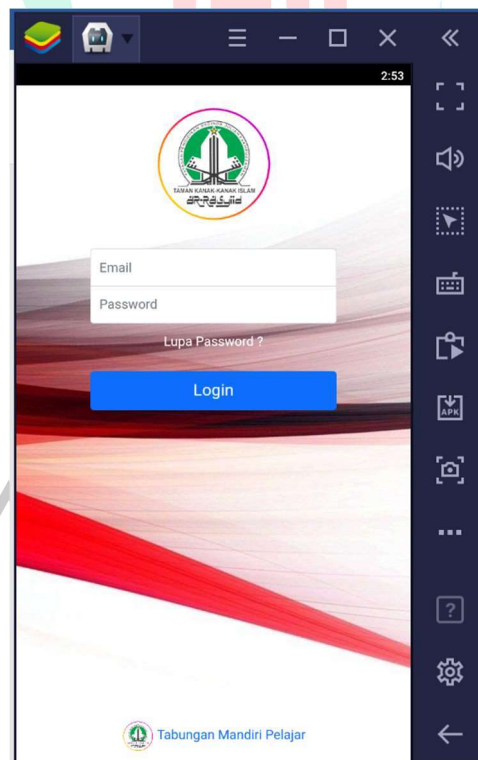
**Gambar 1- 58.** Implementasi Halaman Utama Wali Murid pada *Smartphone*



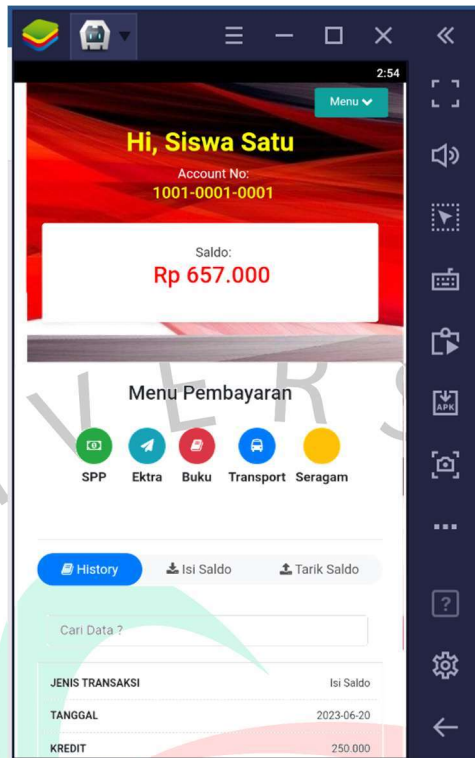
**Gambar 1- 59.** Implementasi Halaman Utama Guru pada *Smartphone*



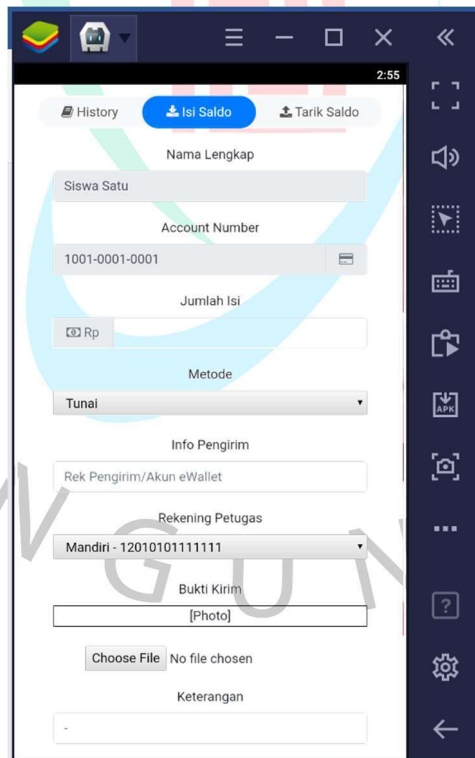
**Gambar 1- 60.** Implementasi Halaman Utama Petugas TU pada *Smartphone*



**Gambar 1- 61.** Implementasi Halaman *Login* pada *Emulator Bluestack*

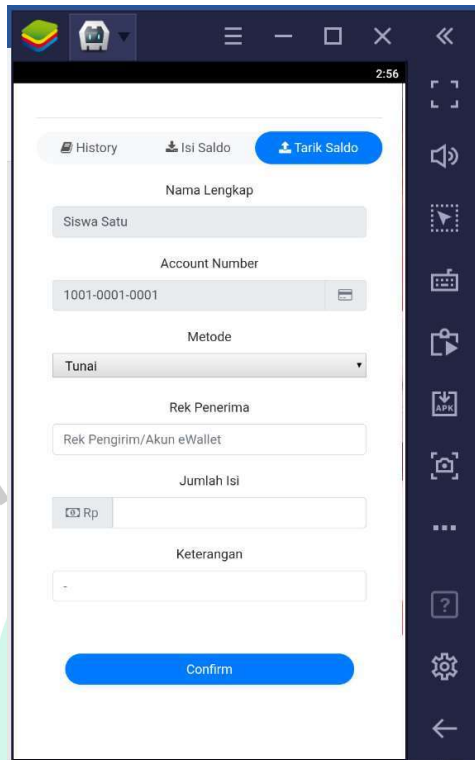


Gambar 1- 62. Implementasi Halaman Utama Wali Murid pada Emulator *Bluestack*

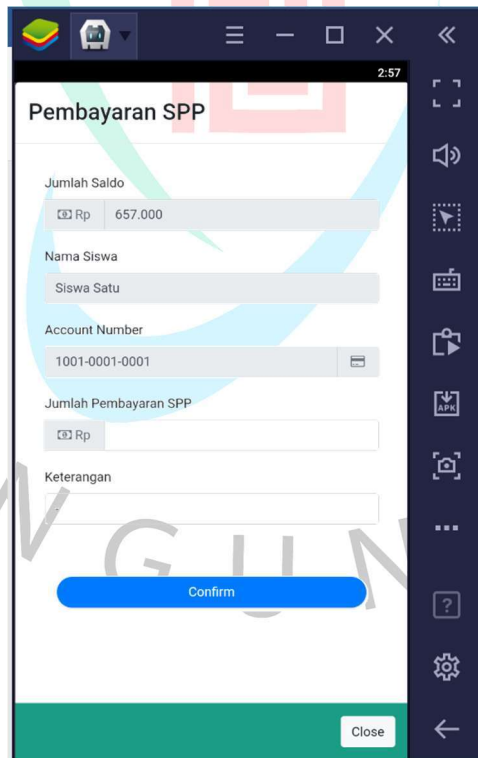


Gambar 1- 63. Implementasi Halaman *Form* Isi Saldo pada Emulator *Bluestack*

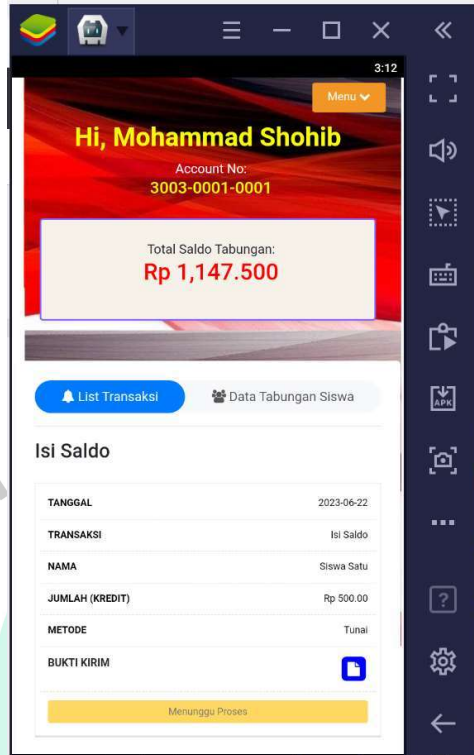




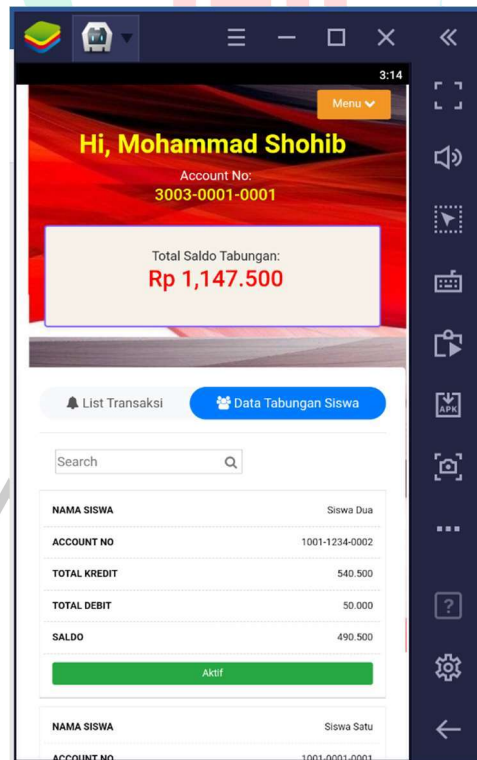
Gambar 1- 64. Implementasi Halaman *Form Tarik Saldo* pada Emulator *Bluestack*



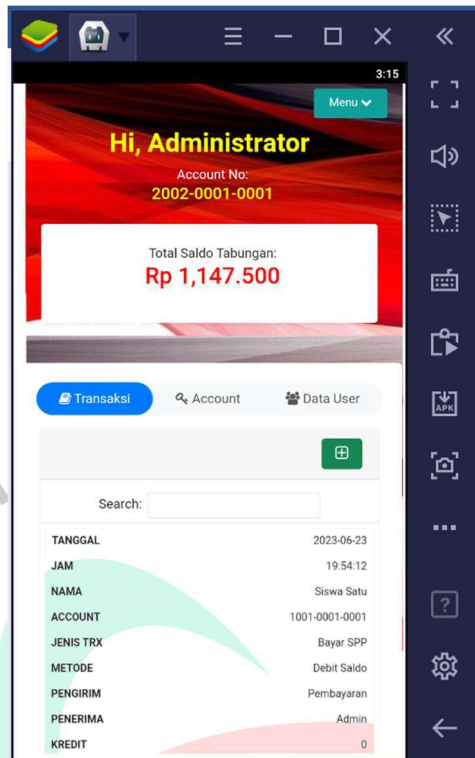
Gambar 1- 65. Implementasi Halaman *Form Pembayaran* pada Emulator *Bluestack*



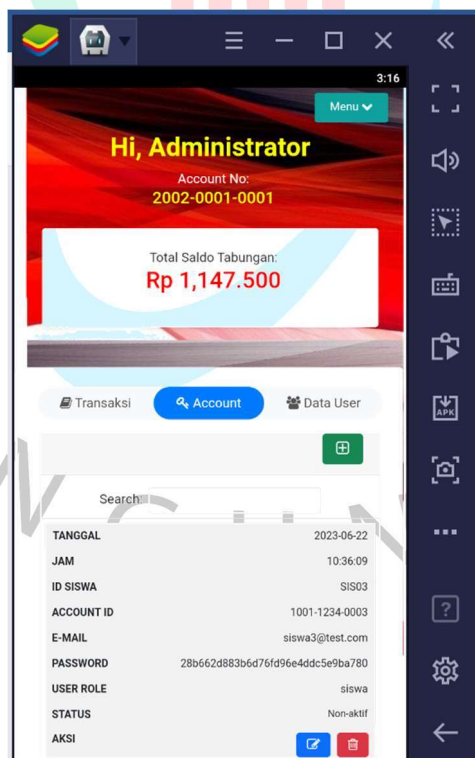
Gambar 1- 66. Implementasi Halaman Utama Guru pada Emulator *Bluestack*



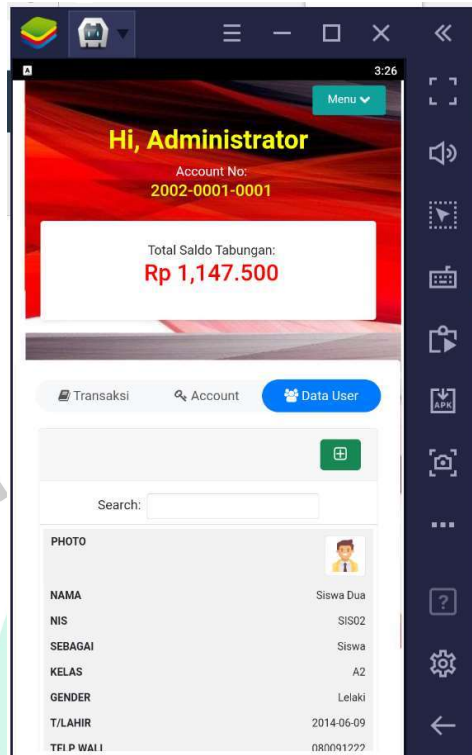
Gambar 1- 67. Implementasi Halaman Data Tabungan Siswa pada Emulator *Bluestack*



Gambar 1- 68. Implementasi Halaman Data Transaksi pada Emulator *Bluestack*



Gambar 1- 69. Implementasi Halaman Data *Account* pada Emulator *Bluestack*



Gambar 1- 70. Implementasi Halaman Data User pada Emulator *Bluestack*

#### 4.6.4. Pengujian Sistem

Pengujian sistem menggunakan *Emulator Android BlueStack* dan *Handphone* berbasis operasi sistem Android dengan metode *Black Box* yang tidak menguji pada *source code* pemograman, pengujian ini menguji terhadap tampilan aplikasi, fungsi-fungsi pada aplikasi serta kesesuaian alur fungsi. Hasil pengujian adalah sebagai berikut:

Tabel 1- 31. Hasil Pengujian

Device	Hasil
<i>Emulator Android BlueStack</i>	Aplikasi dapat terinstall.
	User dapat Login.
	Login akan mengarahkan halaman sesuai level Siswa/Guru/Admin.
	Aplikasi dapat beroperasi.
	Aplikasi dapat di gunakan dari tampilan history, isi saldo, tarik tunai serta melakukan pembayaran.
	Proses kelola data dapat dilakukan dengan baik.
<i>Handphone (Android)</i>	Aplikasi dapat terinstall.
	User dapat Login.

Login akan mengarahkan dalaman sesuai level Siswa/Guru/Admin.
Aplikasi dapat beroperasi.
Aplikasi dapat di gunakan dari tampilan <i>history</i> , isi saldo, tarik tunai serta melakukan pembayaran.
Proses kelola data dapat dilakukan dengan baik.

