

Aplikasi Tabungan Dan Pinjaman Berbasis Akuntansi Perbankan

Nisa Nur Pradipta¹, Magdalena Karismariyanti²

¹Program Studi D3 Sistem Informasi Akuntansi, Fakultas Ilmu Terapan Universitas Telkom

[1nisapradipta22@gmail.com](mailto:nisapradipta22@gmail.com), [2magdalena@tass.telkomuniversity.ac.id](mailto:magdalenatass@tass.telkomuniversity.ac.id)

Keywords:

Mini Bank, web based application, SDLC

Abstract

This research produces a web-based application that can manage savings, loan transactions, installment payments and interest, generate mutations from each customer and financial statements. Banking accounting used follows the standards of the Indonesian Banking Accounting Guidelines (PAPI). The object of research case studies still uses manual recording so that the duration of service takes a long time and often the difference between physical money and manual recording. System Development Life Cycle (SDLC) with the Waterfall model used in building applications. Use Case Testing is done to test the application output according to its function obtained based on thirty five (35) use cases with a test status of 100% success. This application can be used as a solution for learning banking accounting in the laboratory in the academic sphere.

Kata Kunci:

Transaksi, aplikasi berbasis web, SDLC

Abstrak

Penelitian ini menghasilkan aplikasi berbasis web yang dapat mengelola transaksi tabungan, pinjaman, pembayaran angsuran beserta bunga, menghasilkan mutasi dari setiap nasabah dan laporan keuangan. Akuntansi perbankan yang digunakan mengikuti standar Pedoman Akuntansi Perbankan Indonesia (PAPI). Obyek studi kasus penelitian adalah Sekolah Menengah Kejuruan (SMK) masih menggunakan pencatatan secara manual sehingga durasi pelayanan memakan waktu lama dan sering terjadi selisib antara uang secara fisik dengan pencatatan manual. System Development Life Cycle (SDLC) dengan model Waterfall digunakan dalam membangun aplikasi. Use Case Testing dilakukan untuk menguji luaran aplikasi sesuai dengan fungsinya diperoleh berdasarkan tiga puluh lima (35) use case dengan status pengujian berhasil 100%. Aplikasi ini dapat digunakan sebagai solusi untuk pembelajaran akuntansi perbankan di laboratorium pada lingkup akademik.

Pendahuluan

Laboratorium perbankan, selanjutnya disebut Bank Mini, digunakan untuk pembelajaran transaksi perbankan untuk siswa atau mahasiswa dari jurusan akuntansi. Siswa selanjutnya disebut nasabah. Transaksi dalam perbankan diantaranya tabungan terdiri dari setoran dan tarikan, pinjaman dan pembayaran angsuran beserta bunga. Berdasarkan dokumen di Sekolah Menengah Kejuruan (SMK) dari obyek penelitian bulan April tahun 2017 saldo mengalami penurunan sebesar 30% sedangkan pada bulan juli tahun 2017 transaksi mengalami kelonjakan sebesar 35%, dan sisanya ialah 35% saldo stabil. Kelonjakan 35% dipengaruhi dengan adanya transaksi setoran dikarenakan banyaknya siswa baru yang menabung. Saldo yang mengalami penurunan sebesar 30% karena banyaknya siswa yang menarik tabungan.

Untuk membuka rekening baru dalam bank mini, diperlukan adanya aktivasi rekening dengan biaya sebesar Rp. 5.000 supaya nasabah dapat melakukan transaksi setoran. Untuk melakukan transaksi tarikan saldo, nasabah tidak dapat menarik semua saldo yang dimiliki tetapi harus disisakan atau adanya saldo yang mengendap sebesar Rp.5.000. Tabungan bukan merupakan satu – satunya fasilitas untuk mengelola uang, namun terdapat fasilitas lain yaitu adanya pinjaman dana [1] [2]. Pinjaman dapat dilakukan oleh nasabah apabila nasabah memiliki saldo dalam rekening. Pengajuan pinjaman dapat dilakukan dengan syarat sebesar 50% dari saldo yang dimiliki dan angsuran dana digunakan untuk mengembalikan pinjaman [1] [2]. Proses pembukaan rekening baru, setoran, tarikan, pinjaman dan angsuran ditulis ke dalam buku penerimaan dan pengeluaran. Dalam melayani satu nasabah, petugas menulis data pada slip setoran atau tarikan, kemudian ditulis kedalam buku penerimaan atau buku pengeluaran. Catatan dalam buku penerimaan atau buku pengeluaran kemudian ditulis ulang kedalam buku tabungan. Sementara itu, seluruh siswa di SMK diwajibkan menjadi nasabah membuka rekening serta menabung per bulan untuk melatih kesadaran pengelolaan keuangan. Pelayanan yang masih manual dengan minimal tiga (3) dokumen per nasabah dan jumlah nasabah yang banyak tiap bulannya mengakibatkan waktu layanan yang lama.

Hasil dari penelitian adalah membuat aplikasi yang dapat membantu mengelola transaksi [3] [1] [2]. Transaksi tabungan dan pinjaman akan ditampilkan otomatis sesuai dengan pencatatan akuntansi yang dimulai dari jurnal, buku besar. Buku besar akan ditampilkan setiap saat sesuai dengan pilihan periode dan akun yang dibutuhkan. Laporan keuangan berupa laporan laba rugi, neraca ditampilkan sesuai akun standar PAPI yang disesuaikan dengan kebutuhan laboratorium [1].

Kerangka Teoritis

a. Akuntansi Perbankan

Terdapat beberapa jenis dalam akuntansi yaitu akuntansi manajemen, akuntansi pemerintah, akuntansi biaya, akuntansi pajak, akuntansi keuangan, akuntansi perbankan dan sebagainya. Secara umum standar Pedoman Akuntansi Perbankan Indonesia (PAPI) mengatur tentang akuntansi perbankan. Akuntansi perbankan ialah sebuah sistem informasi yang mengolah data dengan cara pencatatan, penggolongan, pengikhtisaran, dan pelaporan yang menghasilkan informasi keuangan sesuai dengan standar yang diakui umum serta digunakan oleh berbagai pihak–pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan [4].

Laporan keuangan dapat digunakan oleh pihak–pihak berkepentingan seperti pemilik perusahaan, kreditur, pemasok, investor, dll. Laporan keuangan berfungsi untuk pengambilan keputusan dalam perusahaan [5]. Dalam pembuatan laporan keuangan harus memiliki kualitas yaitu: relevan dimana laporan keuangan harus sesuai dengan kebutuhan dari pengguna sehingga dapat bermanfaat dan fokus terhadap tujuan umum perusahaan, dapat dimengerti, daya uji yaitu laporan keuangan harus dapat di pertanggung jawabkan akan kebenarannya, netral yaitu laporan keuangan ditunjukkan kepada tujuan umum pengguna tidak hanya kepada pihak – pihak tertentu atau tidak boleh berpihak pada satu pihak saja, tepat waktu, daya banding, lengkap [6].

b. Siklus Akuntansi Perbankan

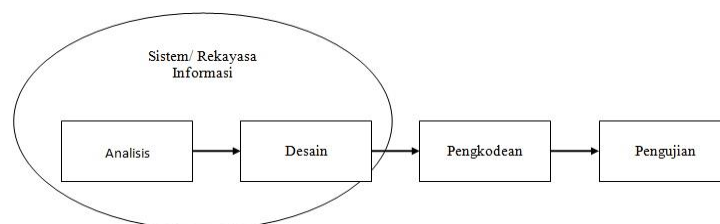
Tahapan dalam akuntansi yang harus dilakukan dalam pencatatan transaksi disebut siklus akuntansi. Siklus akuntansi sangat diperlukan dalam akuntansi perbankan karena berfungsi sebagai urutan dan termasuk tahapan yang sudah ditentukan dalam Pedoman Akuntansi Perbankan Indonesia (PAPI). Dalam melakukan pencatatan transaksi dimulai dari dokumen dasar atau bukti transaksi seperti slip setoran, slip tarikan, cek, *bilyet* giro dan sebagainya yang dijadikan dasar oleh seorang akuntan untuk melakukan pencatatan. Bukti transaksi tersebut dicatat kedalam jurnal pada tabel 1 kemudian di posting kedalam buku besar [7] [8] sehingga menghasilkan neraca saldo atau neraca percobaan yang berisi seluruh daftar akun yang terdaftar dalam buku besar beserta saldo akhir suatu periode akuntansi tertentu [9]. Tahap selanjutnya ialah akun beserta saldo akhir akan disesuaikan saldonya dan dicatat ke dalam jurnal penyesuaian dan dicatat ke dalam neraca lajur/*worksheet*. Neraca lajur ini digunakan hanya sebagai alat untuk memudahkan dalam penggabungan akun beserta saldo setelah adanya penyesuaian dalam jurnal penyesuaian dan bukan keharusan yang harus dibuat sebelum pembuatan laporan keuangan. Jika saldo telah disesuaikan maka dibuat laporan keuangan yang berisi neraca, laporan laba rugi, dan laporan perubahan modal. Setelah tahap pembuatan laporan keuangan selesai, maka tahap selanjutnya ialah membuat jurnal penutup ini berfungsi untuk menutup akun dan saldo sebagai tanda berakhirnya pencatatan dan pelaporan akuntansi untuk periode tertentu. Setelah membuat jurnal penutup akun dan saldo akhir di posting ke dalam buku besar dan membuat neraca penutup. Selanjutnya ialah membuat jurnal pembalik di awal periode yang digunakan untuk mengoreksi jika terjadi kesalahan atau kekeliruan dalam pencatatan dalam menyesuaikan akun di jurnal penyesuaian [10].

Table 1 Jurnal Umum

No	Transaksi	Jurnal Umum				
		Tanggal	Keterangan	Ref	Debet	Kredit
1	Modal awal untuk membuka Bank Mini	01-Jan-18	Kas	111	Rp.50.000.000	
			Modal Disetor	301		Rp. 50.000.000
2	Aktivasi rekening tabungan untuk siswa baru	01-Jan-18	Kas	111	Rp. 500.000	
			Pendapatan Administrasi	402		Rp. 500.000
3	Setoran tabungan	01-Jan-18	Kas	111	Rp. 15.000.000	
			Tabungan	201		Rp. 15.000.000
4	Tarikan tabungan	05-Jan-18	Tabungan	201	Rp. 5.000.000	
			Kas	111		Rp. 5.000.000
5	Pencairan pinjaman	15-Jan-18	Kredit yang diberikan	113	Rp. 25.251.000	
			Kas	111		Rp. 25.251.000
6	Pencatatan Bunga	30-Jan-18	Piutang Bunga	112	Rp. 13.750.000	
			Pendapatan Bunga	401		Rp. 13.750.000
7	Pembayaran angsuran bunga dan pinjaman pokok	30-Jan-18	Kas	111	Rp. 39.001.000	
			Kredit Yang Diberikan	113		Rp. 25.251.000
			Piutang Bunga	112		Rp.13.750.000
8	Pelunasan pinjaman	30-Jan-18	Kas	111	Rp. 3.000.000	
			Kredit Yang Diberikan	113		Rp. 3.000.000

Metode

Metode yang digunakan dalam pengerjaan penelitian ini dengan menggunakan metode *Waterfall* pada tahap *System Development Life Cycle* (SDLC). Model *Waterfall* merupakan salah satu model yang digunakan untuk mengembangkan suatu sistem dalam SDLC, versi *Pressman* memiliki alur yang secara sistematis dengan diawali analisis, desain, pengkodean, pengujian [11]. Tahapannya dapat dilihat pada gambar 1.



Gambar 1 Tahapan SDLC

a. Analisis

Pada tahap ini membutuhkan data yang digunakan untuk merancang perangkat lunak sehingga mudah dipahami. Dalam menganalisis kebutuhan menggunakan *Unified Modeling Language (UML) use case diagram, Entity Relationship Diagram*. Objek penelitian adalah sebuah Sekolah Menengah Kejuruan (SMK) swasta yang berada di kabupaten Bandung. Teknik yang digunakan dalam pengumpulan data adalah wawancara, observasi dan *study literature*. Wawancara kepada dua narasumber yaitu sebagai kepala jurusan dan wakil kepala jurusan dilakukan pada tanggal 28 agustus 2018. Observasi untuk melihat secara langsung proses yang sedang berjalan di bank mini dilakukan selama tiga bulan dari bulan Agustus sampai dengan Oktober tahun 2018. Untuk mendukung kesesuaian hasil penelitian, *study literature* dilakukan dengan telaah referensi berupa buku teks, jurnal dan sumber lain.

b. Desain

Setelah tahap analisis, maka dilakukan tahap desain yang sesuai dengan kebutuhan dari tempat obyek penelitian menggunakan *Balsamiq Mockup*.

c. Pengodean

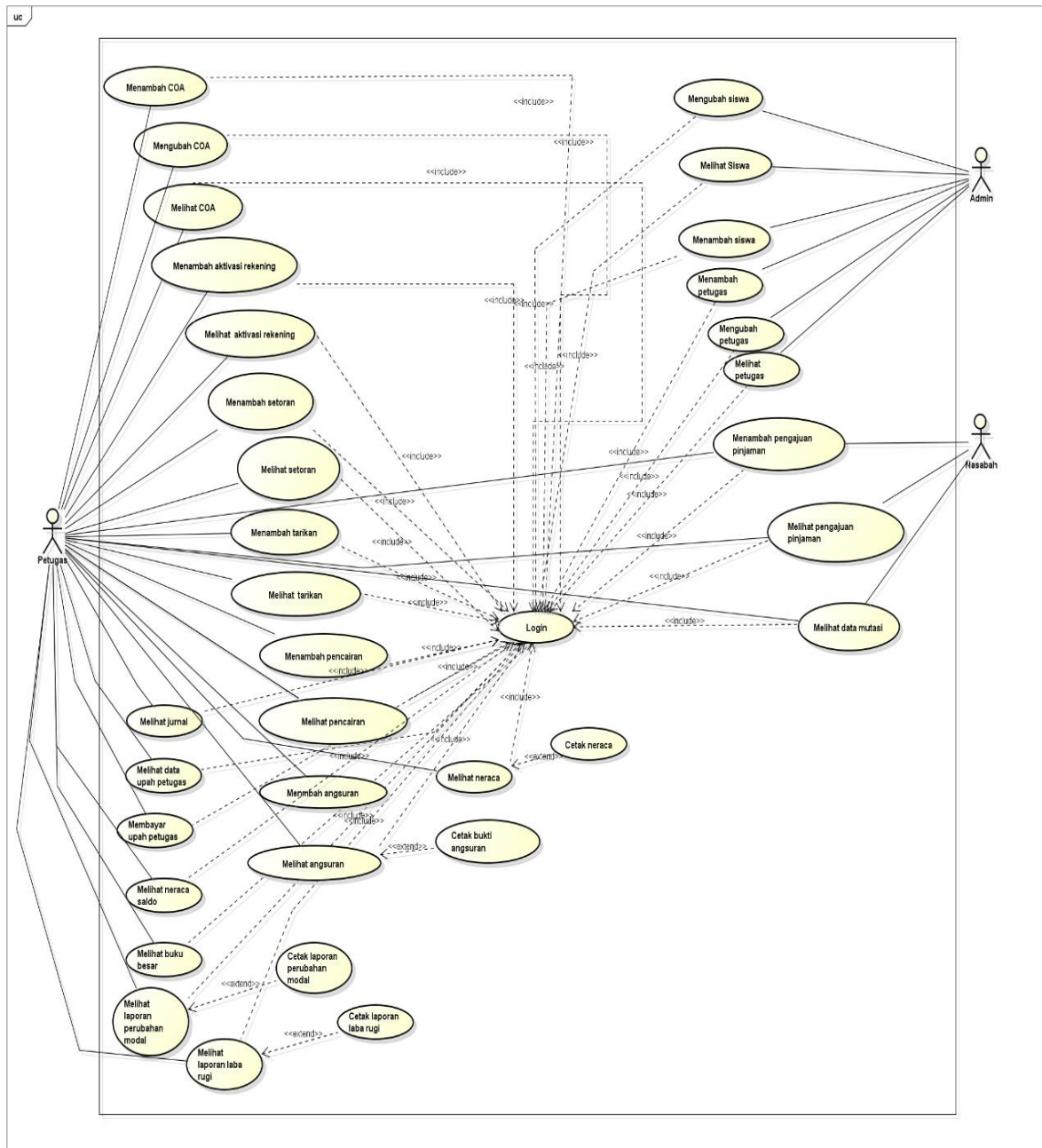
Pengodean dengan pembuatan aplikasi menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter* [12] dan penyimpanan data menggunakan MySQL yang membantu dalam pembuatan perangkat lunak untuk membuat database secara gratis karena termasuk lisensi *Free Software*, sedangkan *Shareware* ialah pengiriman serta penerimaan *database* yang cepat [13].

d. Pengujian

Setelah perangkat lunak selesai, pengujian dilakukan secara sistematis dan sudah direncanakan untuk menguji perangkat lunak baik secara teknis ataupun non teknis. Pengujian berguna meminimalisir kesalahan dalam kinerja perangkat lunak atau pesan *error* yang muncul dalam aplikasi sebelum diberikan kepada pengguna dengan menggunakan *use case testing* yang menguji fungsional dari aplikasi [14].

HASIL DAN PEMBAHASAN

A. Use case Diagram



Gambar 2 Use case Diagram

Use case diagram digunakan untuk menganalisis sistem yang menggambarkan adanya interaksi antara sistem dengan aktor [11] [15]. Terdapat tiga aktor dalam diagram use case yaitu admin, nasabah dan petugas. Jumlah use case dalam use case diagram sebanyak tiga puluh lima (35) sehingga aplikasi ini dapat mengelola transaksi yang terjadi di bank mini.

B. Skenario Use Case

Setiap use case perlu dibuat skenario use case, dalam bagian ini ditampilkan hanya empat (3) skenario yaitu menambah setoran, menambah tarikan, dan menambah pengajuan pinjaman.

1. Menambah Setoran

Table 1 Menambah Setoran

<i>Use Case</i>	: Menambah setoran	
Aktor	: Petugas	
<i>Pre-condition</i>	: Aktor telah <i>login</i> dan berada di halaman menu transaksi sub bab setor/tarik	
<i>Post-condition</i>	: Sistem menyimpan data transaksi setoran ke dalam basis data	
<i>Description</i>	: Proses dimana aktor menambah transaksi setoran	
Aktor		System
Skenario Normal : Data Masukan Valid		
1. Memasukkan data setoran		
		2. Sistem menyimpan data ke dalam <i>database</i>
		3. Sistem kembali pada halaman awal
Skenario Alternatif : Data Masukan Invalid		
		2a. Sistem tidak menerima nilai dan menampilkan pesan <i>error</i> dan meminta aktor untuk mengisi ulang.
		2b. Menampilkan halaman <i>form</i> setoran

2. Menambah Tarikan

Table 2 Menambah Tarikan

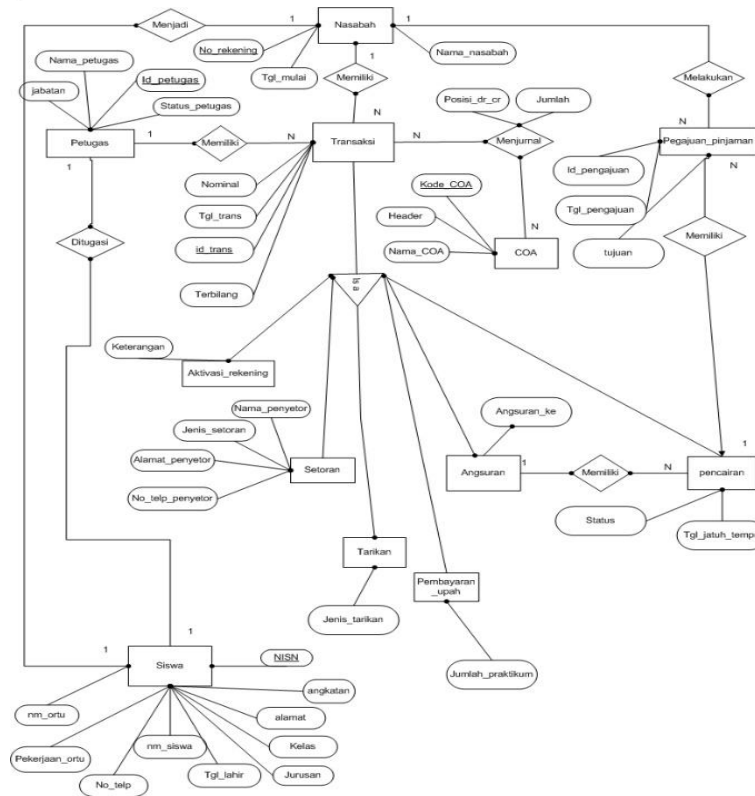
<i>Use Case</i>	: Menambah tarikan	
Aktor	: Petugas	
<i>Pre-condition</i>	: Aktor telah <i>login</i> dan berada di halaman menu transaksi sub bab setor/tarik	
<i>Post-condition</i>	: Sistem menyimpan data transaksi tarikan ke dalam basis data	
<i>Description</i>	: Proses dimana aktor menambah transaksi tarikan	
Aktor		System
Skenario Normal :		
1. Pilih menu transaksi sub bab setor/ tarik		
2. Menekan tombol tambah		
		3. Sistem menampilkan <i>form</i>
4. Mengisi <i>form</i>		
		5. Sistem menyimpan data ke dalam <i>database</i>
		6. Menampilkan data transaksi tarikan yang sudah berhasil

3. Menambah Pengajuan Pinjaman

Table 3 Menambah Pengajuan Pinjaman

<i>Use Case</i>	: Menambah pengajuan pinjaman	
Aktor	: Petugas dan nasabah	
<i>Pre-condition</i>	: Aktor telah <i>login</i> dan berada di halaman menu transaksi sub pinjaman	
<i>Post-condition</i>	: Sistem menyimpan data pengajuan pinjaman ke dalam basis data	
<i>Description</i>	: Proses untuk aktor menambah pengajuan pinjaman	
Aktor		System
Skenario Normal :		
1. Pilih menu transaksi sub bab pinjaman		
2. Menekan tombol tambah		
		3. Sistem menampilkan <i>form</i>
4. Mengisi <i>form</i>		
		5. Sistem menyimpan data ke dalam <i>database</i>
		6. Menampilkan data pengajuan pinjaman yang sudah berhasil

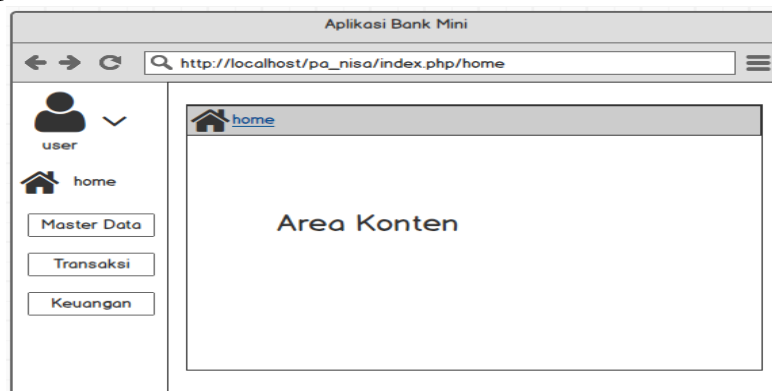
C. Entity Relationship Diagram (ERD)



Gambar 3 Entity Relationship Diagram

Pada saat melakukan perancangan suatu perangkat lunak perlu melakukan pemodelan, *Entity Relationship Diagram* ialah pemodelan paling awal yang digunakan untuk memodelkan basis data [16] [11]. Berdasarkan *Entity Relationship Diagram* (ERD) di atas terdapat dua belas (12) entitas dan satu (1) kardinalitas m-n sehingga menghasilkan tiga belas (13) tabel. Pada saat transaksi setoran dilakukan, tabel yang terpengaruh adalah tabel setoran, transaksi, *Chart Of Account* (COA) dan jurnal.

D. Balsamiq Mockup



Gambar 4 Desain Antarmuka

Desain antarmuka pada penelitian ini terdiri dari *master data*, transaksi, keuangan. *User* ialah orang yang akan *login* ke dalam aplikasi menggunakan *account* yang telah dimiliki. Setelah *user login*, aplikasi akan menampilkan halaman utama yaitu *home*. Dalam menu utama atau yang disebut *home* terdapat menu yang lain terdiri dari *master data*, transaksi, keuangan.

E. Implementasi

Implementasi dilakukan sesuai dengan gambar 1 tahap *System Development Life Cycle* (SDLC) yaitu tahap pengodean. Pengodean pada aplikasi menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) dengan *framework CodeIgniter*.

Gambar 5 Menambah Setoran

Gambar 5 menggambarkan implementasi menambah setoran, *use case diagram* dan skenario *use case* dengan nama menambah setoran. Aktor sebagai petugas menambah setoran dengan memasukkan jumlah setoran. Secara akuntansi dicatat ke dalam jurnal terhadap akun kas di sisi debit dan akun tabungan di sisi kredit, pada tabel 1 nomor 3.

Laporan Laba Rugi		
Tahun : 2019		
Bulan : Oktober		
Pendapatan		
Pendapatan Bunga	Rp 3.278	
Pendapatan Adm	Rp 10.000	
Pendapatan		Rp 13.278
Beban Operasional		
Beban Upah	Rp 0	
Beban Operasional		Rp 0
Laba/Rugi		Rp 13.278

Gambar 6 Laporan Laba Rugi

Gambar 6 menggambarkan implementasi aktor melihat laporan laba rugi sesuai dengan *use case diagram* dengan nama melihat laporan laba rugi. Data dalam laporan laba rugi akan ditampilkan secara otomatis berdasarkan data tabel jurnal pada gambar 3.

Laporan Neraca	
Tahun : 2019	
Bulan : Oktober	
Aktiva	
Kas	Rp 8.897.359
Piutang Bunga	Rp 86
Kredit Yang Diberikan	Rp 3.409.081
Pasiva	
Tabungan	Rp 475.000
Ekuitas	
Modal Disetor	Rp 5.000.000
Saldo Laba Rugi	Rp 13.278
Total Aktiva, Pasiva, Ekuitas	Rp 17.794.804

Gambar 7 Neraca

Gambar 7 menggambarkan implementasi aktor melihat neraca sesuai dengan *use case diagram* melihat neraca. Data aktiva, pasiva, dan ekuitas secara otomatis akan ditampilkan berdasarkan implementasi *use case* menambah aktivasi rekening, melihat aktivasi rekening, menambah setoran, melihat setoran, menambah tarikan, melihat tarikan, menambah pengajuan pinjaman, melihat pengajuan pinjaman, menambah pencairan, melihat pencairan, menambah angsuran dan melihat angsuran. Saldo laba rugi dengan nominal Rp.13.278 pada gambar 6 secara otomatis ditampilkan pada laporan neraca gambar 7.

F. Pengujian

Masukkan data pada *use case testing* ini tidak dapat diturunkan langsung dari *use case* [17, p. Chp 5] [18, p. Chp. 9] maupun *use case scenario*. Teknik penentuan masukkan data dapat menggunakan *equivalence class and boundary value techniques* [18, p. Chp. 9]. Dua langkah awal dalam membuat *test case* dengan *use case testing* [18, p. Chp. 9]: dimulai dengan data normal untuk transaksi sesuai *scenario use case*, kemudian beralih ke nilai batas (*boundary values*) dengan data tidak valid.

Setiap skenario *use case* akan menjadi dasar pembuatan *Test Case*. Ada dengan tiga puluh lima (35) *use case* diuji skenario *use case testing* dengan status pengujian berhasil 100%. Pada table 1 skenario *use case* menjadi *input* data kolom (3) pada table 5. Tabel 5 merupakan *Test Case* dari skenario *use case* pada tabel 1 dengan nama *use case* menambah setoran. Antarmuka aplikasi yang diuji sesuai dengan gambar 5. Pada gambar 5 kolom nomor rekening dan nama nasabah otomatis tampil sesuai dengan pilihan yang dipilih lain halnya dengan kolom jumlah setoran, kolom ini perlu dimasukkan oleh petugas, sehingga kolom ini perlu diuji sesuai kasus pengujian valid dan tidak valid.

Table 4 Pengujian Fungsionalitas Setoran

Test No (1)	Test Case (2)	Input Data (3)	Result		Status (6)
			Expected (4)	Actual (5)	
1.0	Menambah Setoran:Skenario normal	1.1 Memasukkan data Setoran valid a. Jumlah Setoran = 5000 (valid)	Sistem menyimpan data ke dalam database dan sistem kembali pada halaman awal	Sistem menerima nilai dan kembali pada halaman awal	Berhasil

Test No (1)	Test Case (2)	Input Data (3)	Result		Status (6)
			Expected (4)	Actual (5)	
		Menekan tombol simpan			
1.1	Menambah Setoran: Skenario alternatif Jumlah Setoran tidak valid	1.1 Memasukkan data dengan Jumlah Setoran tidak valid a. Jumlah Setoran = 4999 (invalid) Menekan tombol simpan	Sistem tidak menerima nilai dan menampilkan pesan <i>error</i> “ Bagian Jumlah harus lebih dari atau sama dengan 5.000” Sistem menampilkan halaman <i>form</i> setoran	Sistem tidak menerima nilai dan menampilkan pesan <i>error</i> “ Bagian Jumlah harus lebih dari atau sama dengan 5.000” Sistem menampilkan halaman <i>form</i> setoran	Berhasil

PENUTUP

Berdasarkan pembahasan hasil penelitian aplikasi mampu menangani transaksi tabungan yang terdiri dari tarikan dan setoran, pinjaman, pembayaran angsuran beserta bunga serta menghasilkan laporan keuangan berupa laporan laba rugi dan neraca sesuai dengan standar akuntansi. Pengembangan aplikasi penelitian ini disarankan *auto debt* pembayaran Sumbangan Pembinaan Pendidikan (SPP), akses untuk nasabah dengan versi *mobile*.

DAFTAR PUSTAKA

- [1] P. F. Zain, M. Karismariyanti and F. Sukmawati, Aplikasi Pengelolaan Tabungan Sampah dan Dana Pinjaman Nasabah di Bank Sampah Bandung Bersinar, Kabupaten Bandung. Proyek akhir tidak dipublikasikan, Bandung: Universitas Telkom, 2016.
- [2] R. R. Garcia, E. W. Yunarso and G. Prabawa, Aplikasi Berbasis Web Bantuan Pinjaman Tanpa Bunga Dengan Sms Gateway Studi Kasus : Yayasan Garlinda. Proyek akhir tidak dipublikasikan, Bandung: Universitas Telkom, 2017.
- [3] J. C. Nisa, A. P. Kurniawan and S. K. Sari, Aplikasi Pembayaran SPP dan Saldo Tabungan Berbasis Web Disertai SMS Gateway (Studi kasus Mts dan MA Mathla’ul Huda). Proyek akhir tidak dipublikasikan, Bandung: Universitas Telkom, 2017.
- [4] A. S. Nugroho, Akuntansi Bank, Jakarta: Bhuana Ilmu Populer, 2018.
- [5] R. Daud Ga, AKUNTANSI DASAR Teori dan Praktik, Yogyakarta: ANDI, 2017.
- [6] Taswan, Akuntansi Perbankan Transaksi dalam valuta Rupiah, III ed., Yogyakarta: UPP STIM YKPN, 2017.
- [7] S. A. Siregar, Belajar Mudah Akuntansi Dasar Volume 1 Perusahaan Jasa, BAO Publisher, 2018.
- [8] Hery, Pengantar Akuntansi Comprehensive Edition Lengkap Dengan Soal Dan Solusinya, Jakarta: PT Grasindo, 2015.

- [9] S. Samryn LM, Pengantar Akuntansi: Buku 1 Metode Akuntansi untuk elemen Laporan Keuangan Diperkaya dengan Perspektif IFRS dan PERBANKAN, Jakarta: Rajawali Pers, 2015.
- [10] S. Bahri, Pengantar Akuntansi Berdasarkan SAK ETAP dan IFRS, Yogyakarta: CV.Andi Offset, 2016.
- [11] A. S. Rosa and M. Shalahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, Bandung: Informatika, 2013.
- [12] Y. Supardi and A. Hermawan, Semua Bisa Menjadi Programmer CodeIgniter Basic, Elex Media Komputindo, 2018.
- [13] T. Radillah, Visual Basic 6.0 Return, Yogyakarta: Deepublish, 2018.
- [14] A. B. Hikmah, D. Supriadi and T. Alawiyah, Cara Cepat Membangun Website dari Nol Studi Kasus: Web Dealer Motor, Yogyakarta: CV ANDI OFFSET, 2015.
- [15] F. H. Utami and Asnawati, Rekayasa Perangkat Lunak, Yogyakarta: Deepublish, 2015.
- [16] S. M. Indrajani, DATABASE Design Case Study All In One Cara praktis belajar perancangan basis data dengan cepat, tepat dan Jelas Analisis Proses bisnis penjualan, pembelian, dan persediaan (pendekatan terstruktur dan berorientasi objek), Jakarta: PT Elex Media Komputindo, 2015.
- [17] H. Schaefer, T. Linz and A. Spillner, Software Testing Foundations, 4th ed., 2014.
- [18] L. Copeland, A Practitioner's Guide to Software Test Design, Artech House, 2004, p. 300.