

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Suci Ratnawati, Nur Aeni Hidayah dan Risa Oktavian, (2014), mengembangkan aplikasi tentang “*Pengembangan Sistem Informasi Akuntansi Syariah (Studi Kasus: BMT Bintaro)*”. Mereka mengembangkan aplikasi dengan menggunakan diagram UML untuk pemodelan aplikasi yang dibangun dengan menggunakan Bahasa pemrograman *Java* dan menggunakan *MySQL* sebagai basis data. Sistem yang dibangun memberikan layanan transaksi pencatatan data anggota dalam dokumentasi juga penawaran, perhitungan bagi hasil dan juga rincian hasil.

Yoni M. Luthfan N, (2009), mengembangkan aplikasi “*Sistem Pengolahan Data Keuangan Terpadu Untuk BMT Bina Artha Madani berbasis Client-Server*”. Dalam pengembangan websitenya, Yoni M. Luthfan N menggunakan metode iterasi dan menggunakan *diagram ERD, DFD* untuk pemodelan aplikasi yang dibangun menggunakan Aplikasi *WampServer 2.0i* yang merupakan bundel aplikasi berisi aplikasi *server Apache Server 2.2.11, PHP 5.3.0*, dan sistem manajemen basis data *MySQL 5.1.36*. Sistem yang dibangun dibagi dalam beberapa jenis pengolahan data yaitu pengolahan data nasabah (simpan dan pinjam), pengolahan data keuangan, serta pengolahan data pengguna. Masing-masing pengolahan data dibagi hak aksesnya ke dalam tiga tingkat pengguna.

Tuwarno, (2011), dalam pengembangan aplikasi “*Sistem Informasi Simpan Pinjam Koperasi Mitra Mandiri Jetak*”, metode yang digunakan adalah *UML (Unified Modelling Language)* untuk perancangan aplikasi. *Software* yang digunakan dalam pengembangan aplikasi menggunakan Bahasa pemrograman *Java* dan *MySQL* sebagai basis data. Sistem yang dibangun memberikan layanan pencatatan transaksi dan laporan dari transaksi simpan pinjam.

Skripsi yang berjudul “*Aplikasi Simpanan berjangka di BMT Marhamah Wonosobo*”, Oleh Muhammad Chanif, (2015). Prosedur dan Simpanan berjangka diawali dengan pengisian form aplikasi Simpanan simka yang diisi oleh deposan. BMT akan memberikan sertifikat simka kepada deposan, Sertifikat ini berfungsi sebagai tanda bukti kepemilikan simpanan di BMT Marhamah yang ditunjukkan kepada BMT.

Dari keempat penelitian sebelumnya, aplikasi desktop yang akan dibuat memberikan layanan untuk pencatatan dan penghitungan transaksi simpan pinjam yang hanya dapat dilakukan oleh pihak manajemen dengan hak akses masing-masing. Berdasarkan penelitian yang sudah ada, maka penulis menambah beberapa spesifikasi yang menjadi perbedaan dengan sistem yang dikembangkan.

Perbedaan tersebut adalah sebagai berikut:

1. Aplikasi dapat mencatat transaksi angsuran pinjaman dalam periode yang sudah ditentukan oleh pihak manajemen.
2. Aplikasi dapat mencatat sumber *infaq* yang diberikan oleh anggota.
3. Aplikasi dapat menampilkan laporan simpanan, pinjaman dan angsuran berdasarkan laporan bulanan.

2.2 Landasan Teori

2.2.1 Produk-produk *Baitul Maal wat Tamwil*

Operasional BMT adalah hampir sama dengan BPR *Syari'ah*. Yang membedakan hanyalah pada sisi lingkup dan struktur. Dilihat dari fungsi pokok operasional BMT, ada dua fungsi pokok dalam kaitannya dengan kegiatan perekonomian masyarakat, kedua fungsi tersebut adalah:

a. Produk pengumpulan dana BMT

1. Simpanan *Wad'iah*

Adalah titipan dana yang tiap waktu dapat ditarik pemilik atau anggota dengan cara mengeluarkan semacam surat berharga pemindah bukuan/transfer dari perintah bayaran lainnya.

2. Simpanan *Mudharabah*

Adalah simpanan pemilik dana yang penyetorannya dan penarikannya dapat dilakukan sesuai dengan perjanjian yang telah disepakati sebelumnya. Adapun jenis-jenis tabungan/simpanan di BMT adalah sebagai berikut:

- a) Tabungan persiapan *qurban*
- b) Tabungan Pendidikan
- c) Tabungan Persiapan untuk nikah
- d) Tabungan persiapan untuk melahirkan
- e) Tabungan naik haji/umroh
- f) Simpanan Berjangka/deposito
- g) Simpanan khusus untuk kelahiran
- h) Simpanan sukarela

b. Produk Pembiayaan

Pembiayaan uang dan tagihan berdasarkan persetujuan atau kesepakatan pinjam meminjam antara BMT dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi utangnya beserta bagi hasil setelah jangka waktu tertentu.

1. Pembiayaan *al-Murabahah* (MBA)
2. Pembiayaan *al-Bai' Bitsaman Ajil* (BBA)
3. Pembiayaan *al-Mudharabah* (MDA)
4. Pembiayaan *al-Musyarakah* (MSA)
5. Pembiayaan *Al-Mudhorobah* (*Turst Financing, Trust Investmen*)

2.2.2 Defnisi Sistem Informasi

Menurut Gelianas et al. (2005) sistem informasi adalah sistem buatan manusia yang secara umum mencakup suatu set komponen berbasis komputer dan komponen manual yang saling terintegrasi yang dibangun untuk mengumpulkan, menyimpan, dan mengelolah data serta untuk menyediakan keluaran berupa informasi bagi para pengguna.

Bodnar dan Hopwood (2000:4) mengungkapkan bahwa istilah sistem informasi menganjurkan penggunaan teknologi komputer di dalam organisasi untuk menyajikan informasi kepada pemakai. Sistem informasi “berbasis komputer” merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat.

2.2.3 Software Development Life Cycle

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta, 2007). Selain itu, SDLC juga penting untuk proses *maintenance software* itu sendiri. Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall*. Menurut Sommerville (2011), *waterfall model* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *ClassicLife Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

2.2.4 Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk

sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.

2.2.5 Alur Proses

2.2.5.1 Alur Proses Simpanan (Baitul Maal wat Tamwil)

Secara garis besar proses Simpanan *Baitul Maal wat Tamwil* terdiri dari 3 proses yang dapat dilihat pada gambar 2.1 :



Gambar 2.1 Proses Simpanan

Berikut adalah penjelasan tentang Gambar 2.1 :

a. Anggota

Setelah terdaftar menjadi anggota dengan melakukan setoran Simpanan wajib, simpanan pokok dan simpanan sukarela. Pada setoran selanjutnya anggota mengisi dan menandatangani slip setoran yang berisikan tanggal, jenis simpanan, nomor rekening, nama pemilik dan jumlah nominal yang akan disimpan atau ditabung.

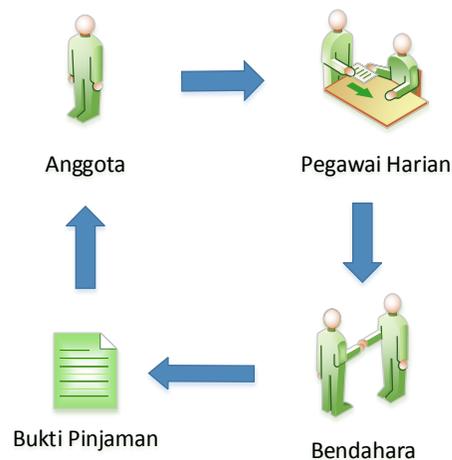
b. Pegawai Harian

Pegawai harian menerima uang tunai dan mencocokkan jumlahnya dengan jumlah yang tertera pada slip setoran. Pegawai harian dapat memvalidasi slip setoran pada printer dan mencetak setoran dan saldo terakhir pada buku simpanan. Kemudian slip setoran dan uang tunai diserahkan kepada bendahara.

c. Bendahara

Bendahara membuat pembukuan dengan slip setoran anggota, dan laporan secara berkala pada setiap bulan.

2.2.5.2 Alur Proses Peminjaman



Gambar 2.2 Proses Pinjaman

Berikut adalah penjelasan tentang Gambar 2.2 :

Proses peminjaman yang diajukan anggota kepada pihak manajemen adalah sebagai berikut:

a. Anggota

Anggota mengajukan permohonan secara tertulis kepada BMT dengan mengisi formulir pengajuan. Proses ini dilakukan oleh pegawai harian BMT. Ini dilakukan setelah semua persyaratan formal dipenuhi, seperti yang menyangkut legalitas calon peminjam (data diri, dsb).

b. Pegawai Harian

Bila seluruh proses oleh pegawai harian telah selesai dilakukan, dokumen yang berisi usulan pembiayaan tersebut diserahkan ke bagian administrasi pembiayaan (bendahara) untuk diperiksa kelengkapannya. Selanjutnya dimintakan persetujuan komite pembiayaan. Setelah usulan pembiayaan tersebut mendapat persetujuan dari

komite pembiayaan, tahap selanjutnya adalah mempersiapkan pengikatan pembiayaan (akad pembiayaan). Sebelum dilakukan pengikatan, semua dokumen asli dan dokumen jaminan harus telah diterima

c. Bendahara

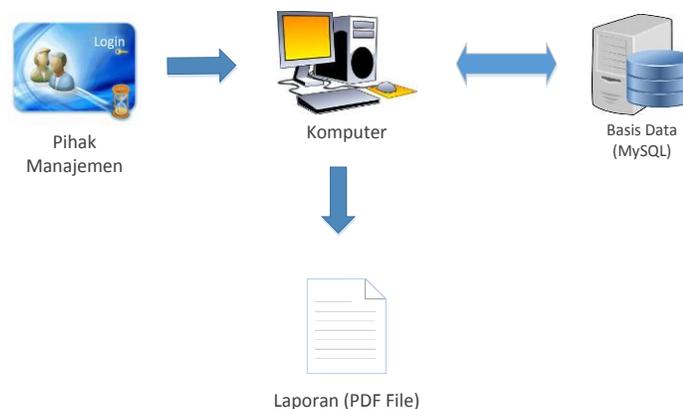
Setelah dilakukan pengikatan pembiayaan, proses pencairan dana dilakukan oleh bendahara, dengan terlebih dahulu dilakukan verifikasi tanda tangan calon peminjan.

d. Bukti Pinjaman

Bukti pinjaman akan dicap menurut jenis pinjamannya, setelah selesai ditandatangani oleh pegawai harian dan bendahara maka akan dikembalikan kepada anggota.

2.2.6 Arsitektur Sistem

Dalam mengembangkan aplikasi diperlukan perancangan arsitektur perangkat lunak yang bertujuan untuk menggambarkan bagaimana sistem dikembangkan dan dijalankan. Arsitektur perangkat lunak pada aplikasi dapat dilihat pada Gambar 2.3.



Gambar 2.3 Arsitektur Sistem

Berikut adalah penjelasan tentang Gambar 2.3 :

a. Pihak Manajemen

Pihak Manajemen merupakan orang yang mengoperasikan aplikasi. User adalah pegawai dari BMT yang biasa melakukan proses pendataan dan transaksi.

b. Komputer

Komputer merupakan sebuah perangkat keras yang digunakan oleh pegawai BMT sebagai alat untuk menjalankan banyak aplikasi. Komputer digunakan saat sedang menjalankan aplikasi. Sebelum dijalankan, aplikasi harus diinstal terlebih dahulu di dalam komputer.

d. Basis Data

Basis data digunakan untuk menampung data-data yang telah dimasukkan oleh pihak manajemen.

e. Laporan

Laporan merupakan hasil (*output*) dari aplikasi. *Report* yang diharapkan dari aplikasi dapat berupa arsip transaksi simpanan, laporan transaksi pinjaman dan angsuran pinjaman yang dapat otomatis diarsipkan pada setiap bulan.

2.2.7 Teknologi Pengembangan Aplikasi

2.2.7.1 United Markup Language (UML)

United Markup Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model UML bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi penjualan dan pembelian antara lain adalah *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*

a. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan

fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol dalam Use Case Diagram

| NO | Gambar | Nama Gambar | Keterangan |
|----|---|-----------------------|--|
| 1 |  | <i>Use Case</i> | Merupakan fungsionalitas yang disediakan system sebagai unit yang bertukar pesan dengan <i>actor</i> . |
| 2 |  | <i>Actor</i> | Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> . |
| 3 |  | <i>Association</i> | Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem. |
| 4 |  | <i>Generalization</i> | Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem. |

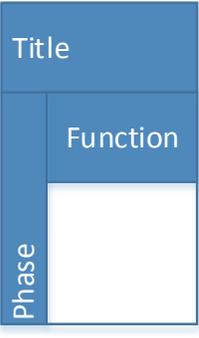
| | | | |
|---|---|----------------|--|
| 5 |  | <i>Include</i> | Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>use case</i> memanggil <i>use case</i> yang lain. |
| 6 |  | <i>Extend</i> | Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi. |

b. Activity Diagram

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol dalam Activity Diagram

| NO | Gambar | Nama Gambar | Keterangan |
|----|---|-----------------------|---|
| 1 |  | <i>Start Point</i> | Merupakan awal dalam aktifitas |
| 2 |  | <i>End Point</i> | Merupakan akhir dalam aktifitas |
| 3 |  | <i>Activities</i> | Menggambarkan suatu proses atau kegiatan bisnis |
| 4 |  | <i>Decision Point</i> | Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas |

| | | | |
|---|---|-----------------|---|
| 5 |  | <i>Swimlane</i> | Digunakan untuk pembagian <i>activity diagram</i> yang menunjukkan siapa yang melakukan aktifitas |
|---|---|-----------------|---|

c. Class Diagram

Class Diagram merupakan diagram yang digunakan untuk menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

2.2.7.2 Basis Data

Basis data merupakan salah satu teknologi yang harus dimiliki sebuah perusahaan, institusi, ataupun organisasi adalah teknologi yang dapat memproses data. Secara konsep basis data atau database adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tata cara tertentu untuk membentuk data baru atau informasi. Kumpulan dari data yang saling berhubungan (*relasi*) antara satu dengan lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Supriyanto, 2005).

Untuk mengelola database diperlukan suatu perangkat lunak yang disebut DBMS (*Data Base Management System*). DBMS merupakan suatu system perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol, dan mengakses database. Dengan DBMS, pengguna dapat mengontrol dan memanipulasi data yang ada.

2.2.7.3 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang di distribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tidak boleh dijadikan produk turunan yang bersifat komersial.

Berikut beberapa keistimewaan dari MySQL menurut beberapa sumber sebagai bahan pertimbangan:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka (*open source*). MySQL didistribusikan sebagai open source sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed or unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti *password* yang terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan *protocol TCP/IP, Unix Soket (UNIX)*, atau *named pipes (NT)*.

10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.7.4 SQLite JDBC

Untuk mengakses data yang ada di dalam database SQLite, maka diperlukan sebuah API yaitu SQLite JDBC. Fungsi dari SQLite JDBC sendiri adalah untuk mengeksekusi statement-statement SQL (seperti mengelompokkan data, menambah sebuah *record*, menciptakan tabel, dan lain sebagainya). JDBC mendukung secara penuh aktifitas untuk:

1. Melakukan koneksi ke sebuah Driver Database Relasional
2. Dengan koneksi yang dibangun, JDBC dapat mengirim statement-statement SQL ke driver database relasional (seperti: *select*, *CRUD*, *resultset*, *metadata*, dan lain-lain).
3. Menampilkan sebuah set tabel dan relasi yang diinginkan.

2.2.7.5 Bahasa Pemrograman Java

Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan

aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".

Penulis memakai bahasa java dalam mengembangkan aplikasi karena java dapat dijalankan di beberapa *platform* atau sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

a. Kelebihan Java

1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*. Dengan kelebihan java, *programmer* cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan java memungkinkan sebuah program berbasis java dikerjakan di atas *operating system* Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung adalah Microsoft Windows, Linux, Mac OS, dan Sun Solaris.
2. Perpustakaan kelas yang lengkap. Java terkenal dengan kelengkapan yang sangat memudahkan dalam penggunaan oleh para *programmer* untuk membangun aplikasinya. Kelengkapan perpustakaan ditambah dengan keberadaan komunitas java yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
3. Memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak programmer C++ untuk pindah ke Java.

b. Kekurangan Java

1. Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Java merupakan *bytecode* yang menyimpan banyak

atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program lebih sulit disembunyikan dan mudah dibajak.

4. Penggunaan memori yang banyak. Biasanya bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berurusan dengan mesin komputer berumur lebih dari 4 tahun.

2.2.6 Perangkat Lunak Pendukung

2.2.6.1 NetBeans

NetBeans merupakan perangkat lunak (*software*) yang digunakan untuk mengembangkan aplikasi, di dalam perangkat lunak terdapat IDE (*Integrated Development Environment*). Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan GUI (*Graphic User Interface*).

2.2.6.2 IReport

IReport merupakan sebuah aplikasi designer yang digunakan untuk membuat laporan yang didalamnya terdapat Library JasperReport. Library JasperReport sendiri merupakan JAR (*Java Library*) yang bersifat open source dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi java. JasperReport memiliki sejumlah fitur, antara lain :

1. Layout dan desain laporan yang fleksibel.
2. Dapat menampilkan laporan dalam bentuk teks maupun gambar (*chart*).
3. Dapat menghasilkan report dalam berbagai format : html, pdf, rtf, xls, csv.
4. Dapat menerima data dari berbagai sumber data : JDBC, Bean Collection, ResultSet, CSV, XML, Hibernate.

2.2.7 Aplikasi

Menurut Anisyah (2000), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian aplikasi, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah Microsoft Office dan OpenOffice.org yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya.

Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2.8 Aplikasi Dekstop

Menurut Konixbam (2009), aplikasi desktop adalah suatu aplikasi yang dapat berjalan sendiri dalam suatu komputer dengan *operating system* atau *platform* tertentu tanpa menggunakan browser atau koneksi Internet. Dalam aplikasi terdapat beberapa halaman yaitu:

a. Halaman Anggota

Halaman anggota diperlukan untuk memasukan dan menampilkan data diri dari anggota, untuk mengetahui nomor dari anggota yang akan digunakan untuk setiap transaksi pada BMT sehingga memudahkan dalam pencarian transaksi.

b. Halaman Menu Simpanan

Halaman transaksi simpanan diperlukan untuk memasukan dan menampilkan data setiap anggota yang melakukan setoran. Tiap simpanan dibedakan dengan kode simpanan SW(simpanan wajib) dan SS(simpanan sukarela).

c. Halaman Menu Pinjaman

Halaman transaksi pinjaman diperlukan untuk memasukan data transaksi pinjaman yang diajukan oleh anggota. Data pinjaman diperlukan agar pihak BMT dapat mengetahui detail nama peminjam, jumlah pinjaman dan lamanya angsuran yang dilakukan anggota, dan pihak manajemen BMT dapat mengetahui pegawai harian yang menerima transaksi pinjaman anggota.

d. Halaman Menu Angsuran

Halaman transaksi angsuran digunakan untuk mencatat transaksi angsuran dari pinjaman anggota, data transaksi mencatat nomer pinjaman, nominal dangsuran dan angsuran ke- yang disetorkan oleh peminjam. Jumlah pinjaman akan otomatis berkurang setelah data angsuran disimpan.

e. Halaman Menu Laporan

Laporan dalam aplikasi dibagi menjadi beberapa bagian yaitu :

1. Laporan simpanan peranggota, untuk mengetahui detail simpanan pokok, simpanan wajib dan simpanan sukarela dari seorang anggota.
2. Laporan simpanan perbulan, untuk mengetahui data transaksi pinjaman yang dilakukan anggot dalam rentang waktu satu bulan. Lporan simpanan perbulan mencatat siapa saja yang melakukan setoran simpanan pada satu bulan tertentu.
3. Laporan pinjaman anggota, untuk mengetahui transaksi pinjaman yang dilakukan oleh anggota. Laporan pinjaman anggota diurutkan berdasar tanggal pinjaman.
4. Laporan angsuran peranggota, untuk mengetahui angsuran pinjaman dari seorang anggota, berapa kali anggota tersebut sudah melakukan angsuran dan jumlah sisa pinjaman yang belum dikembalikan.
5. Laporan angsuran perbulan, untuk mengetahui data transaksi angsuran pada setiap bulan. Pada setiap bulan dapat dilakukan pengecekan siapa saja yang melakukan setoran angsuran dan nama pegawai harian yang menerima angsuran dari anggota.