

# SISTEM INFORMASI PENGELOLAAN JAMINAN KREDIT PADA BPR MCM MENGGUNAKAN ARSITEKTUR N-TIER

Rubi Tridarta Sandiraga<sup>1</sup>, Sujito<sup>2</sup>, Weda Adistianaya Dewa<sup>3</sup>  
Program Studi Sistem Informasi, STMIK Pradnya Paramita Malang  
[rubitsandiraga@gmail.com](mailto:rubitsandiraga@gmail.com)<sup>1</sup>, [sujito@stimata.ac.id](mailto:sujito@stimata.ac.id)<sup>2</sup>, [weda@stimata.ac.id](mailto:weda@stimata.ac.id)<sup>3</sup>

## Abstract

*PT. BPR Mitra Catur Mandiri (BPR MCM) is one of the Rural Bank serves the provision of credit, savings and deposits to customers both for personal use and development of micro enterprises. PT. BPR Mitra Catur Mandiri also provide facilities to the debtor to replace the collateral. Recording replace the current collateral using book of registers and application that named SOLegal where the application is not able to record the history of data collateral. In addition, the Credit Division and Legal Division annoyed the demand list of installments and collateral the debtor in exchange for collateral process. The method applied in this study is the method of application architecture and system development methods. Method of application architecture used is N-Tier Data Application web based. While the system development method used is the System Development Life Cycle (SDLC) to the stages of software requirement analysis, design, program code generation, testing, support or maintenance. The result showed that the use of N-Tier architecture, management information systems collateral can be accessed from the central office and branch office and be able to record the history of the initial collateral when credit disbursement and after replacing the collateral.*

**Keyword:** BPR, Management of Collateral, N-Tier Data Application, ASP.Net, Web

## Abstrak

PT. BPR Mitra Catur Mandiri (BPR MCM) adalah salah satu Bank Pedesaan yang melayani penyediaan kredit, tabungan, dan deposito kepada nasabah baik untuk penggunaan pribadi maupun pengembangan usaha mikro. PT. BPR Mitra Catur Mandiri juga menyediakan fasilitas bagi debitur untuk mengganti agunan. Pencatatan penggantian agunan saat ini menggunakan buku register dan aplikasi bernama SOLegal, di mana aplikasi tersebut tidak mampu mencatat riwayat data agunan. Selain itu, Divisi Kredit dan Divisi Hukum merasa terganggu dengan daftar tuntutan angsuran dan agunan debitur dalam proses penggantian agunan. Metode yang diterapkan dalam penelitian ini adalah metode arsitektur aplikasi dan metode pengembangan sistem. Metode arsitektur aplikasi yang digunakan adalah Aplikasi Data N-Tier berbasis web. Sedangkan metode pengembangan sistem yang digunakan adalah Siklus Hidup Pengembangan Sistem (SDLC) dengan tahapan analisis kebutuhan perangkat lunak, desain, pembuatan kode program, pengujian, dukungan atau pemeliharaan. Hasil penelitian menunjukkan bahwa dengan menggunakan arsitektur N-Tier, sistem informasi manajemen agunan dapat diakses dari kantor pusat dan kantor cabang serta mampu mencatat riwayat agunan awal saat pencairan kredit dan setelah penggantian agunan.

**Kata kunci:** BPR, Manajemen Jaminan, Aplikasi Data N-Tier, ASP.Net, Web

## PENDAHULUAN

PT. BPR Mitra Catur Mandiri (BPR MCM) adalah salah satu Bank Perkreditan Rakyat yang melayani pemberian fasilitas kredit, tabungan dan deposito kepada para nasabah baik untuk keperluan pribadi maupun pengembangan usaha mikro. Hingga akhir bulan Oktober 2014, perusahaan ini memiliki pendapatan Rp.5.549.143.959,92. Pendapatan tersebut sebagian besar berasal dari fasilitas kredit dengan jumlah Rp.5.190.016.086,31 dengan 1.650 debitur aktif dan 4.960 rekening tabungan.

PT. BPR Mitra Catur Mandiri memiliki 3 cabang yang tersebar di Malang Raya, antara lain Kantor Kas Turen, Kantor Kas Kasembon, Kantor

Kas Ngebruk dan satu Kantor pusat yang berdiri di Jalan Raya Pakis No. 2B, Pakis, Malang.

Kantor pusat merupakan pusat dari aktivitas manajemen PT BPR Mitra Catur Mandiri. Segala aktivitas mulai dari pendaftaran nasabah baru, permohonan kredit, pendaftaran tabungan, pengelolaan jaminan, pencairan kredit hingga proses akuntansi dan laporan pada Bank Indonesia dilakukan di kantor pusat.

Kantor kas merupakan cabang dari kantor pusat. Kantor kas ini berfungsi sebagai fasilitas untuk memperluas pasar dan memudahkan debitur yang berlokasi jauh dari kantor pusat dalam pengajuan kredit maupun tabungan serta mengangsur kredit.

Setiap sore hari kantor kas akan mengirimkan data transaksi setoran kredit, tabungan, dan pencairan kredit berupa fax pada kantor pusat. Kemudian kantor pusat akan memasukkan data tersebut pada sistem. Dalam pelayanan kredit, calon debitur mengajukan permohonan kredit beserta data jaminan (agunan) kepada pihak bank kemudian akan diproses oleh Divisi Legal yang bertugas untuk membuat dan mengesahkan perjanjian kredit. Kemudian, debitur menyerahkan jaminan berupa sertifikat tanah atau Bukti Pemilik Kendaraan Bermotor (BPKB) kepada Divisi Legal yang selanjutnya jaminan tersebut disimpan pada tempat penyimpanan khusus dan dicatat dalam buku registrasi dan program aplikasi sederhana yang berbasis desktop yang bernama SOLegal.

Aplikasi SOLegal digunakan untuk mencatat data debitur, data jaminan yang debitur jaminkan pada bank, serta apakah debitur tersebut pernah mengganti jaminan di tengah-tengah masa pinjaman. Namun aplikasi SOLegal hanya mampu mencatat jaminan yang ada saat ini dan tidak mencatat riwayat penggantian jaminan.

Buku registrasi, digunakan untuk mencatat keluar masuknya jaminan setiap harinya dan mencatat nomor surat perjanjian kredit namun tidak mencatat jaminan apa saja yang dijaminkan oleh debitur.

PT. BPR Mitra Catur Mandiri juga memberikan fasilitas kepada debitur untuk mengganti jaminan yang nilai taksasinya lebih kecil apabila debitur telah mengangsur 80 persen dari total pinjaman. Selain itu, debitur juga dapat mengganti jaminan dengan nilai taksasinya yang lebih besar jika debitur tersebut akan menambah fasilitas kredit. Namun fasilitas tersebut membuat Divisi yang terkait merasa tidak nyaman karena dalam prosesnya, pekerjaan Divisi Kredit menjadi terganggu. Hal tersebut terjadi karena sistem ini masih memiliki kelemahan.

Kelemahan yang dimaksud adalah proses penggantian jaminan tidak langsung dilakukan oleh Divisi Legal, namun dimulai dari Marketing yang membuat form pengajuan penggantian jaminan dan meminta daftar angsuran pada Divisi Kredit disela-sela pekerjaan yang selanjutnya diserahkan pada Direksi. Setelah Direksi menyetujui, form tersebut dikembalikan pada Marketing kemudian diserahkan pada Divisi Legal. Divisi Legal akan melakukan validasi form pengajuan tersebut dengan memeriksa data jaminan, sisa angsuran dan calon jaminan pengganti. Disamping itu, pihak kantor kas yang membutuhkan data yang sama harus menunggu 1-2 hari dengan datang langsung ke kantor pusat. Dengan adanya hal tersebut membuat proses pelayanan debitur semakin lama karena debitur harus menunggu 2 hari untuk mengganti jaminan. Disamping itu, Direksi yang sering tidak ada di kantor membuat proses penggantian memakan waktu hingga 4 hari. Proses tersebut terbilang cukup lama jika dibandingkan proses pergantian jaminan di kantor pusat yang hanya membutuhkan 1-2 hari.

Permasalahan berikutnya pun terjadi pada saat Divisi Legal melakukan stok opname jaminan. Divisi Legal harus menyamakan data antara aplikasi dan brankas. Karena aplikasi tersebut berbasis desktop dan hanya bisa dijalankan di komputer lokal, Divisi Legal harus menginstal aplikasi tersebut di notebook kemudian data hasil stok opname ditindih ke data yang ada pada komputer lokal. Hal ini sangat beresiko jika Divisi Legal belum selesai melakukan stok opname karena data yang ada pada komputer lokal akan tertindih dan membuat data menjadi kacau.

Aplikasi yang sifatnya lokal ini tidak mampu mencatat jaminan apa saja yang pernah dijaminkan kepada bank dan tidak mampu mencatat apakah jaminan tersebut digunakan untuk 2 perjanjian kredit secara bersamaan atau tidak.

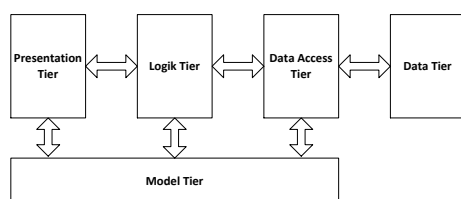
Dari permasalahan tersebut di atas dibutuhkan sistem pengelolaan jaminan yang mampu mencatat riwayat jaminan, mengurangi tugas Divisi Kredit dalam mencetak list angsuran serta menentukan apakah calon jaminan pengganti layak menggantikan jaminan yang sudah ada.

*N-Tier Data Applications* merupakan aplikasi yang dipisahkan menjadi beberapa tingkatan. *N-Tier Data Application* disebut juga aplikasi terdistribusi atau aplikasi *multi-tier (multi-layer)*. *N-Tier Applications* memisahkan proses ke dalam beberapa tingkatan yang didistribusikan antara *client* dan *server*.

Konsep *N-Tier Data Applications* secara umum terbagi menjadi tiga bagian, antara lain *Presentation Tier*, *Middle Tier* dan *Data Tier*. *Presentation Tier* berfungsi sebagai *interface* yang berhadapan langsung dengan *user*. *Middle Tier* berfungsi jembatan yang menghubungkan antara *Presentation Tier* dengan *Data Tier*. *Data input* yang dimasukkan pada *Presentation Tier* akan diolah dan dikirim melalui *Middle Tier* kemudian disimpan pada *Data Tier*. *Data Tier* merupakan lapisan dimana data yang diinputkan pada *Presentation Tier* disimpan.

Teknologi *N-Tier Data Applications* yang bersifat *client-server* dan pemisahan proses menjadi beberapa tingkat memungkinkan pengembangan sistem pengelolaan jaminan bisa diakses dari kantor pusat maupun kantor kas dan memberikan beberapa pilihan untuk pengembangan aplikasi tanpa mengubah struktur inti dari aplikasi tersebut.

Sistem informasi pengelolaan jaminan akan dibangun dengan konsep *5-Tier (Layer)* yang dijelaskan pada gambar 1 berikut.



Gambar 1. Konsep 5-Tier Sistem Pengelolaan Jaminan

*Presentation Tier* merupakan *user interface* yang berhadapan langsung dengan *user* melalui *web browser*. Sebagai contoh, *user* menginputkan data jenis jaminan. Data jenis jaminan yang diinputkan ke dalam *user interface* akan dikirim pada *Model Tier* menjadi sebuah *object* jenis jaminan. *Presentation Tier* akan memanggil fungsi pada *Logic Tier*. Fungsi tersebut akan memanggil *object* jenis jaminan pada *Model Tier*. Kemudian *Logic Tier* akan memanggil fungsi *Query Database* dan mengirimkan *object* jenis jaminan ke *Data Access Tier*. Kemudian *Data Access Tier* akan menjalankan fungsi sesuai perintah *Logic Tier* baik untuk input data maupun mengambil data ke *Data Tier* yang merupakan *database* sistem informasi pengelolaan jaminan.

Keunggulan teknologi *N-Tier* terletak pada kemudahan pengembangan aplikasi yang tidak perlu merubah keseluruhan struktur aplikasi pada saat pengembangan. Pada saat akan mengembangkan aplikasi, pengembang tidak perlu merubah struktur aplikasi, jika ingin mengembangkan *interface* atau memindahkan pada *platform* lain, maka yang perlu diubah adalah *Presentation Tier*. Selain itu jika terjadi perubahan aturan bisnis perusahaan, maka yang perlu diubah hanyalah *Logic Tier*. Perubahan *database* juga menjadi lebih mudah karena hanya perlu mengubah *Data Tier* tanpa harus mengubah lapisan yang lain. Ditinjau dari segi keamanan aplikasi, *N-Tier Data Applications* yang terdiri beberapa lapisan akan lebih sulit ditembus karena harus melewati beberapa tahapan lapisan untuk mencapai lapisan data yang disimpan dalam *database*.

Arsitektur *N-Tier Data Application* akan digunakan untuk membangun sistem informasi pengelolaan jaminan agar bisa mengintegrasikan proses antara kantor pusat dan kantor kas sehingga

dengan sistem ini, pengguna baik Divisi Marketing tidak perlu datang ke kantor pusat untuk melakukan penggantian jaminan dan data riwayat jaminan akan tercatat tanpa ada data yang tertindih dengan data yang baru.

### **Jaminan**

Pato (2013:877) menyatakan bahwa “Jaminan kredit diartikan sebagai penyerahan kekayaan atau pernyataan kesanggupan seseorang untuk menanggung pembayaran kembali suatu utang. Jaminan pemberian kredit diperoleh melalui penilaian yang seksama terhadap watak, kemampuan membayar, modal, dan prospek usaha debitur.”

Keberadaan jaminan (*collateral*) dalam bidang perkreditan dipandang penting meski tidak dapat dikatakan mutlak. Memang pada prinsipnya tidak selalu pengucuran kredit oleh bank harus disertai dengan adanya syarat jaminan, sebab jaminan sudah dianggap ada dengan melihat peluang dan prospek usaha yang baik (prospektif) calon debitur.

### **N-Tier**

Santika (2010:202) mendefinisikan N-Tier adalah pengembangan dari model jaringan *client server* lebih dari 2 tier. Istilah menunjukkan banyaknya lapisan dalam sebuah aplikasi. Sebuah aplikasi terdiri dari beberapa komponen utama, yaitu lapisan *presentation (Presentation Layer)*, lapisan aplikasi (*Application Layer*), atau lapisan logika bisnis (*Business Logic Layer*), dan lapisan data (*Data Layer*). Lapisan *presentation* berhubungan dengan antarmuka dengan pengguna aplikasi, dapat berupa model grafis atau pun berbasis teks.

Pendapat berbeda disampaikan situs resmi [msdn.microsoft.com](http://msdn.microsoft.com) yang mendefinisikan N-Tier sebagai berikut.

*A typical n-tier application includes a presentation tier, a middle tier, and a data tier. The easiest way to separate the various tiers in an n-tier application*

*is to create discrete projects for each tier that you want to include in your application. For example, the presentation tier might be a Windows Forms application, whereas the data access logic might be a class library located in the middle tier. Additionally, the presentation layer might communicate with the data access logic in the middle tier through a service such as a service. Separating application components into separate tiers increases the maintainability and scalability of the application. It does this by enabling easier adoption of new technologies that can be applied to a single tier without the requirement to redesign the whole solution. In addition, n-tier applications typically store sensitive information in the middle-tier, which maintains isolation from the presentation tier.*

Dari kedua definisi tersebut dapat disimpulkan bahwa N-Tier merupakan arsitektur aplikasi yang terdiri dari *n layer* dimana *n* menunjukkan level yang dimiliki aplikasi. Secara umum aplikasi N-Tier dibagi menjadi 3 bagian, yaitu *Presentation Tier*, *Middle Tier* dan *Data Tier*.

### **Presentation Tier**

Noviyanto (2011:482) berpendapat bahwa “Presentation tier tersimpan pada komputer client. Peletakkan bagian ini di bagian client mempercepat proses eksekusi perintah pengguna terhadap aplikasi. Presentation Tier ini merupakan bagian interface aplikasi yang digunakan untuk menampilkan data, komponen dan menangani respon dari request pengguna.”

*Presentation Tier* merupakan tingkat di mana pengguna berinteraksi dengan aplikasi. Terkadang dalam tingkatan ini ditambahkan logika aplikasi namun tidak banyak. *Presentation Tier* tidak secara langsung mengakses *Data Tier*. *Presentation Tier* berkomunikasi dengan *Data Tier* dengan cara mengakses komponen pada *Middle tier*.

### **Middle Tier**

Noviyanto (2011:482) mendefinisikan *Middle Tier* sebagai berikut.

*Middle Tier* berfungsi untuk mengakses dan memproses pengelolaan data. *Middle Tier* menjadi perantara transfer data antara bagian data tier dan presentation tier. Keluaran data yang dipanggil dari data tier ditampilkan dalam format XML, sehingga data akan berada di sisi client. Hal tersebut untuk menangani request data secara realtime, karena pengaksesan data tidak memerlukan komunikasi dengan server secara berulang-ulang.

*Middle Tier* merupakan tempat dimana *Presentation Tier* berkomunikasi dengan *Data Tier*. *Middle Tier* merupakan inti dari aplikasi. Lapisan ini merupakan tempat pengolahan data, validasi dan logika bisnis seperti perhitungan yang kompleks serta *query database*. *Middle Tier* biasanya akan dikembangkan lagi menjadi beberapa lapisan dengan alasan *independency* dan *security*.

Gill (2014:1019) membagi *Middle Tier* menjadi 3 bagian sebagai berikut.

#### 1. **Client presenter layer.**

*This layer consists of the presentation logic required by clients, such as ASP .NET MVC in IIS web server. It also adapts different clients to the business layer.*

#### 2. **Business layer.**

*This layer is also known as domain layer because it handles and encapsulates all the business logics and domains.*

#### 3. **Persistence layer.**

*This is known as data access layer (DAL) because it handles the read/write of data to data layer.*

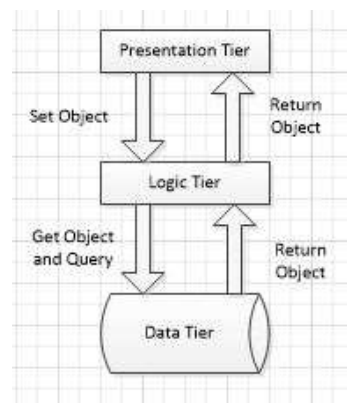
Dari pembahasan di atas dapat disimpulkan bahwa *Middle Tier* dapat dimodifikasi atau dikembangkan dengan sesuai gaya pemrograman pengembang sistem informasi karena tujuan utama

dari pimisahan lapisan adalah agar pada saat pengembangan sistem tidak merubah struktur keseluruhan aplikasi.

#### **Data Tier**

Noviyanto (2011:482) menyatakan bahwa “*Data tier* merupakan database aplikasi yang berada di server. Peletakkan bagian tersebut di server dikarenakan pengelolaan data yang terpusat. Bagian ini sebagai tempat penyimpanan data yang akan ditampilkan dan diolah pada bagian presentasi.”

Pada dasarnya *Data Tier* adalah server yang menyimpan data aplikasi (misalnya, server yang menjalankan SQL Server). *Data Tier* tidak dapat diakses secara langsung dari klien dalam *Presentation Tier*. Sebaliknya, komponen akses yang ada di *Middle Tier* digunakan untuk komunikasi antara *Presentation Tier* dan *Data Tier*.



Gambar 2. Konsep Arsitektur N-Tier

#### **Keuntungan N-Tier Data Applications**

Menurut Gill (2014:1019) keuntungan yang diperoleh menggunakan N-Tier adalah sebagai berikut.

##### **(1) Scalable**

*This is due to its capability of deployment of multiple tiers and the tier decoupling it brought. For example, with the help of database clustering data tier can be scaled up without affecting other tiers.*

##### **(2) Better fault tolerance ability**

*The N-Tier architecture has an ability of fault tolerance. For example, for load balance purpose, databases can be clustered without affecting the other layers.*

### **(3) Independent tier upgrading and changing without affecting other tiers**

*Interface dependency implementation can decouples all layers very well in the object-oriented world, so that, each layer can change individually without affecting the other layers too much.*

### **(4) Friendly and efficient for development**

*The decoupled layers are very software development friendly and efficient. Each layer can be assigned to a team individually, who experts in the specific functional area; an expert team can handle the relevant functional area better and more efficiently.*

### **(5) Friendly for maintenance**

*The N-Tier architecture do the grouping of different things together according to the functionality and then makes things easily understandable, clear and manageable.*

### **(6) Friendly for new feature addition**

*N-Tier architecture has the capability to add new features to the system without affecting it due to the logical grouping of components.*

### **(7) Better reusability**

*Due to the loose couplings among layers, reusability can be done well. In general ways, loosely-coupled component groups implemented, so they can be reused by other applications.*

Keuntungan arsitektur aplikasi *N-Tier* adalah mampu digunakan berulang kali dengan hanya mengubah lapisan presentasi, adanya kemudahan untuk menambah fitur baru tanpa membongkar isi aplikasi, kemudahan dalam perawatan, bila terjadi pembaruan sistem pada salah 1 lapisan tidak akan memngubah keseluruhan sistem, toleransi kesalahan akan mudah terdeteksi jika salah 1 lapisan mengalami *error* dan kemudahan untuk

melakukan *cluster database* tanpa mengganggu lapisan lain.

### **Kerugian N-Tier Data Applications**

Gill (2014:1020) juga menambahkan kerugian arsitektur *N-Tier* sebagai berikut.

(1) *Since the tiers are physically separated, they must communicate across the machine boundaries, process boundaries. This results in high communications overhead.*

(2) *Software installation and up gradation costs and other administration costs are high.*

(3) *It's complicated to design and model.*

(4) *If the network bandwidth and hardware aren't good enough, the performance of an application may be slow.*

*N-Tier* juga memiliki kekurangan, yaitu sulitnya untuk merancang desain dan model sistem, dibutuhkan *bandwith* yang besar dan biaya mahal.

### **UML**

Rosa. A. S (2013:133) mendefinisikan "UML adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek."

### **METODE**

#### **Metode Pengumpulan Data**

Adapun teknik pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut.

#### **(1) Wawancara**

Melakukan wawancara terkait permasalahan yang terjadi pada sistem yang berjalan saat ini.

#### **(2) Observasi**

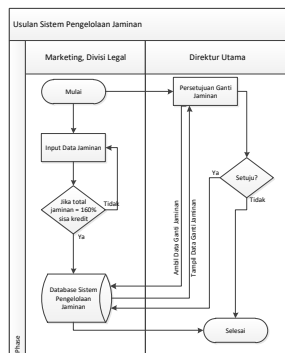
Melakukan analisis secara langsung sistem yang berjalan dan mekanisme sistem yang berjalan saat ini.

#### **(3) Studi Kepustakaan**

Mengumpulkan referensi terkait dari buku dan jurnal penelitian relevan yang mampu mendukung pengembangan sistem.

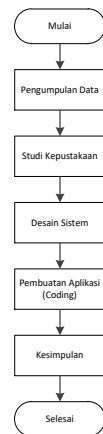
### Sistem yang Diusulkan

Gambaran umum sistem yang diusulkan merupakan tindak lanjut dari analisa sistem pengelolaan jaminan yang berjalan saat ini dan analisa kebutuhan fungsional sistem yang dirancang. Sistem yang diusulkan akan dibangun dengan konsep *client-server* berbasis web sehingga sistem tersebut dapat diakses dari kantor pusat dan kantor kas.



Gambar 3. Flowchart Sistem Usulan

### Tahapan Penelitian



Gambar 4. Tahapan Penelitian

Tahap pertama dimulai dengan pengumpulan data. Pengumpulan data meliputi wawancara dengan divisi terkait terutama Divisi Legal dan Divisi Kredit melakukan observasi proses pergantian jaminan dan input data jaminan.

Tahap penelitian pun dilanjutkan dengan studi kepustakaan dan jurnal penelitian. Studi

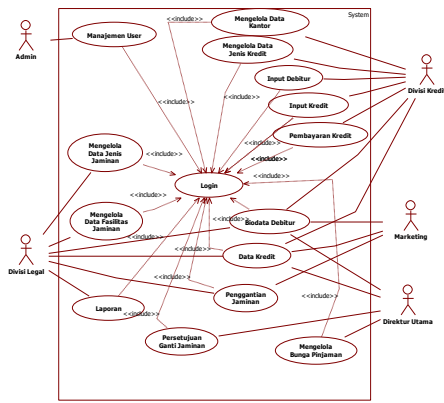
kepustakaan dan jurnal penelitian bertujuan untuk mendapatkan referensi terkait sistem yang dibangun.

Proses dilanjutkan dengan desain sistem. Desain sistem meliputi analisis data dan permasalahan yang terjadi pada sistem penggantian jaminan yang berjalan saat ini dan merancang arsitektur sistem, yaitu *N-Tier Data Application Class Diagram, Use Case Diagram, Activity Diagram, Sequence Diagram, Conceptual Data Model, Physical Data Model*, membangun *Database* dan merancang *Desain Interface*.

*C Tier. Model Tier* digunakan untuk mendeskripsikan mendeskripsikan *class-class* dan menampung hasil inputan dari *Presentation Tier*. Semua inputan ini akan diproses di *Logik Tier* yang kemudian akan dikirim ke *Data Access Tier*. Sederhananya *Model Tier* merupakan kumpulan atribut yang ada pada setiap *class*. Setelah itu input (*object*) yang ada di *Model Tier* akan dikirim ke *Data Access Tier*. *Data Access Tier* berisi *setting koneksi database* dan *query database*. Pada saat *query* tersebut dijalankan *Data Access Tier* akan memanggil *Model Tier* yang kemudian akan diinputkan ke *Data Tier* yang merupakan *database* aplikasi itu sendiri yang terletak pada *server*.

### Use Case Diagram

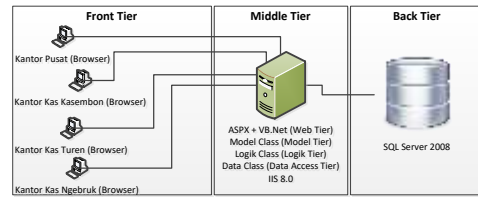
Model *Use Case Diagram* akan terbentuk setelah melakukan indentifikasi aktor dan tugasnya yang langsung berkaitan dengan sistem yang dibangun. Hasil dari dari indentifikasi aktor adalah *Use Case Diagram* yang disajikan pada gambar 6 berikut.



Gambar 6. Use Case Diagram

### Implementasi N-Tier Data Applications

Setiap komputer terhubung dari kantor pusat dan kantor kas terhadap server yang ada pada kantor pusat (*Middle Tier*). Aplikasi sistem pengelolaan jaminan diletakkan pada server yang dijalankan dengan menggunakan *Internet Information Service* (IIS) versi 8.0. Pada aplikasi pengelolaan jaminan ini terdapat 4 lapisan *project* yang terdiri dari *Data Access Tier*, *Logik Tier*, *Model Tier* dan *Web Tier* serta 1 lapisan pada *back tier* yaitu lapisan data yang terdiri dari *database SQL Server 2008*. *Data Access Tier* bertugas untuk mengelola *query* dan koneksi *database* ke *database SQL Server 2008*. *Logik Tier* bertugas sebagai jembatan data yang diinputkan oleh *Web Tier* dan pusat *business logic*. *Model Tier* bertugas mengelola atribut setiap *class* yang ada pada aplikasi ini. Kemudian *Web Tier* merupakan sebagai tampilan dari aplikasi yang akan disajikan pada *browser* ketika aplikasi ini diakses oleh pengguna dari komputer masing-masing. Setiap pengguna akan diberikan hak akses terhadap aplikasi sesuai dengan tugas masing-masing.



Gambar 7. Implementasi Arsitektur N-Tier Data Application

Dengan menggunakan *N-Tier Data Applications* membuat aplikasi ini lebih terstruktur dan memiliki tingkat keamanan yang lebih baik dari sisi aplikasi karena pada saat sistem berjalan, proses akan saling terbagi dan pada setiap lapisannya. Selain itu ketiga lapisan baik *Data Access Tier*, *Logik Tier*, dan *Model Tier* dicompile menjadi file *Dynamic Link Library (.dll)* sehingga aliran data menjadi aman.

Dari sisi jaringan, karena sistem ini diakses menggunakan *Virtual Private Network (VPN)*. Teknologi VPN memungkinkan aplikasi dapat diakses dari kantor kas dan pusat menggunakan jaringan internet secara aman. Teknologi mampu mengenkripsi dan mendekripsi data yang dikirim dari pihak *client* menuju *server* dan sebaliknya.

*Desain Interface* pada sistem pengelolaan jaminan ini menggunakan *Cascading Style Sheet (CSS) Bootstrap* sehingga tampilan dapat menyesuaikan dengan lebar layar saat diakses dari *notebook*, *Personal Computer (PC)*, dan *smartphone*.

### Hasil Penilaian Sistem Informasi Pengelolaan Jaminan

Hasil penilaian sistem informasi pengelolaan jaminan yang telah diujicobakan pada tempat penelitian. Uji coba dilakukan bertujuan untuk mengetahui apakah sistem tersebut sesuai dengan kebutuhan pengguna.

Pengumpulan data uji coba menggunakan angket dengan memberikan 8 daftar pertanyaan seputar tampilan, kemudahan dan kelengkapan informasi. Daftar pertanyaan tersebut disebarakan pada 10 pengguna yang berbeda.

Hasil angket tersebut diolah menggunakan skala *Likert*. Budiaji, Weksi (2013:126) berpendapat bahwa “Skala *Likert* menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan, sangat setuju, setuju, tidak memutuskan, tidak setuju dan sangat tidak setuju. Kemudahan penggunaan skala *likert* menyebabkan skala ini lebih banyak digunakan oleh peneliti.”

Tabel 1 merupakan hasil penilaian sistem informasi pengelolaan jaminan dengan menggunakan skala *likert*.

Tabel 1. Hasil Perhitungan Skala *Likert*

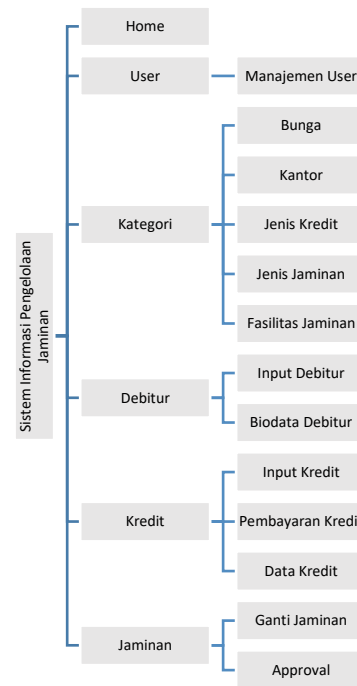
No	Interval	Nilai	Ket.
1.	0 - 19,99%		Sangat Kurang
2.	20 - 39,99%		Kurang
3.	40 - 59,99%		Cukup
4.	60 - 79,99%	79%	Baik
5.	80 - 100%		Sangat Baik

Dari uji coba hasil perhitungan tersebut di atas dapat disimpulkan bahwa sistem informasi pengelolaan jaminan yang diujicobakan pada pengguna sudah sesuai dengan kebutuhan pengguna.

### Site Map

*Site Map* atau peta situs merupakan struktur denah yang berisi menu-menu yang ada dalam website dalam bentuk tautan link yang saling terkait. Sitemap berguna sebagai sebuah informasi tentang url-url yang terdapat dalam sebuah web/blog kepada search engine. Dengan begitu akan mempermudah search engine untuk mengetahui apa saja yang berada dalam sebuah web/blog.

Berikut adalah *site map* website sistem informasi pengelolaan jaminan.



Gambar 8. *Site Map* Sistem Informasi Pengelolaan Jaminan

## KESIMPULAN DAN SARAN

Berdasarkan pembahasan di atas dapat disimpulkan bahwa dengan menggunakan arsitektur *N-Tier* sistem informasi pengelolaan jaminan dapat diakses dari kantor pusat dan kantor kas serta mampu mencatat riwayat jaminan baik jaminan awal pencairan kredit maupun setelah mengganti jaminan. Sistem informasi tersebut merupakan aplikasi *multiuser* yang dapat berjalan optimal pada browser *Google Chrome* dan mampu menyesuaikan tampilan dengan lebar layar perangkat. Berdasarkan hasil angket penilaian terhadap kesesuaian kebutuhan dengan 10 responden dihasilkan nilai 79% dengan kategori baik. Saran untuk penelitian selanjutnya bisa mengembangkan sistem informasi ini dengan metode yang berbeda dan menguji dengan website yang berbeda.

## REFERENSI

AL-Mukhtar, Mumtaz., Hadi, Sarmad. 2012 . *Developing a Three-Tier Web Data Management Application for Higher Education Admission Environment*. *International Arab Journal of e-Technology*, 2 (4): 175-180.

- Anggraeni, Nova., Retnadi, Eko., Kurniawati, Rina. 2012. *Perancangan Sistem Informasi Simpan Pinjam Di KUD Mandiri Bayongbong*. Jurnal Algoritma, 9 (05): 1-11.
- A.S, Rosa., Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Penerbit Informatika
- Budiaji, Weksi. 2013. *Skala Pengukuran Dan Jumlah Respon Skala Likert*. Jurnal Ilmu Pertanian Perikanan, 2 (2): 125-131.
- Effendi, Diana., Wiratama, I. Made 2012. *Perancangan Aplikasi Sistem Pelacakan Software Berlisensi dan Help Desk System (Studi Kasus : PT. Pupuk Kujang Cikampek Jawa Barat)*. Jurnal Penelitian Pos dan Informatika, 2 (1): 51-62.
- Gill, Amandeep. Kaur., Singh, Charanjit. 2014. *Security of N-Tier Architecture using NTRU*. International Journal of Advanced Research in Computer Science and Software Engineering, 4 (7): 1018-1022.
- Ladjamudin, Al-Bahra. Bin. 2013. *Analisis dan Desain Sistem Informasi (Cetakan Kedua)*. Yogyakarta: Graha Ilmu.
- Noviyanto, Fiftin., Ashari, Ahmad. 2011. *Penerapan Teknologi RIA Untuk Membangun Aplikasi Web Dengan Pengaksesan Realtime (Studi Kasus: Pemesanan Handphone Online)*. Jurnal Informatika, 5 (2): 478-490.
- Pato, Saduldyn. 2013. *Analisis Pemberian Kredit Mikro Pada Bank Syariah Mandiri Cabang Manado*. Jurnal EMBA, 1 (4): 875-885.
- Prasanth, Y., Uttam, V. Gowthami., Anusha, G., Priya, G. Sindhu. 2009. *Identifying a generic N-Tier architecture framework for web-based applications*. Oriental Journal of Computer Science & Technology, 2 (2): 153-160.
- Santika, I. Made Wira Adi., Binanto, Iwan. 2010. *Layanan Informasi Untuk Pasien Rumah Sakit Menggunakan SMS*. Jurnal Penelitian, 13 (2): 199-216.
- Williams, Barry. 2012. *Databases Answers Data Modelling by Example*. London: Database Answers Ltd.