

RANCANG BANGUN SISTEM INFORMASI TAKSASI AGUNAN MANAGEMENT RISIKO BISNIS PADA BANK ABC

(Design Of Business Risk Management Collateral Taxation Information System At Bank ABC)

Dewi Cahyanti Aprilia^[1], Royana Afwani^[1], I Wayan Agus Arimbawa^[1]

^[1]Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia

Email: dewiaprilia2001@gmail.com, [royana, arimbawa]@unram.ac.id

Abstract

Lembaga keuangan merupakan instrument penting dalam menunjang pembangunan nasional yang berfungsi sebagai *financial intermediation* dari pihak yang membutuhkan dana. Kegiatan bank pada dasarnya untuk mendapatkan profit tidak jarang memiliki risiko wanprestasi. Risiko ini mampu dihadapi dengan adanya jaminan atau agunan berupa barang bernilai atau non-barang sebagai instrument pengikat nasabah. Bank memiliki layanan agunan yang dikelola oleh Manajemen Risiko Bisnis dalam melakukan pencatatan data agunan secara konvensional menggunakan excel oleh analis. Analis akan meng-input data, dicetak dan diserahkan kepada supervisor guna approval secara manual. *Human error* kerap dihadapi dikarenakan jumlah data yang besar. Pencarian data juga tergolong sulit karena excel tidak menyimpan agunan nasabah yang telah melalui *approval* maupun *rejection*. *Approval* atau *rejection* sulit dilakukan karena data hanya dapat diakses secara lokal dan harus dilakukan *update* secara berkala. Penelitian ini merancang aplikasi berbasis website guna *access mobility* yang mudah dengan bahasa pemrograman PHP dan *framework laravel* dan model pengembangan sistem *Waterfall*. Pengujian fungsionalitas fitur menggunakan metode *black box testing* berjalan dengan baik tanpa adanya *error* dan hasil pengujian *Mean Opinion Scores* (MOS) untuk mengukur kualitas kinerja sistem menghasilkan rata-rata persentase sebesar 95,69%. Sehingga Sistem Informasi Taksasi Agunan Manajemen Risiko Bisnis pada Bank ABC layak dan dapat untuk digunakan serta diselesaikan sesuai jadwal yang direncanakan.

1. PENDAHULUAN

1.1. Latar Belakang

Lembaga keuangan menjadi inti perekonomian suatu negara yang sangat berperan dalam mendorong kemajuan pembangunan negara tersebut. Dalam perannya sebagai perantara keuangan, lembaga keuangan berfungsi sebagai jembatan antara individu atau entitas yang membutuhkan dana dengan yang menyediakan dana. Di Indonesia, lembaga keuangan yang tergabung dalam sistem perbankan dapat dibedakan menjadi dua kategori, yaitu bank konvensional dan bank syariah. Kedua jenis bank ini mengumpulkan dana dari masyarakat dalam bentuk simpanan, kemudian menyalurkannya kembali melalui pemberian kredit atau layanan lain dalam proses pembayaran[1].

Salah satu bentuk tindakan operasional yang dilakukan bank dalam penyaluran dana kepada nasabah adalah memberikan pinjaman dana dengan konsekuensi

mendapatkan keuntungan. Namun, selain keuntungan juga terdapat risiko wanprestasi yang tidak dapat dipungkiri dan disebabkan banyak faktor seperti usaha nasabah macet, keadaan di mana debitur tidak sanggup memenuhi kewajiban yang ditentukan dalam sebuah perjanjian karena keterlambatan pelunasan angsuran tidak disebabkan oleh faktor eksternal atau keadaan yang memaksa [2]. Sehingga apabila kuantitas nasabah semakin tinggi setiap tahunnya, maka semakin besar pula risiko kredit macet dan menyebabkan kekhawatiran tersendiri untuk aktifis perbankan. Untuk mengatasi situasi tersebut, solusi yang dapat diterapkan adalah mewajibkan setiap nasabah untuk memberikan jaminan berupa aset berharga, baik dalam bentuk barang maupun non-barang, atau surat-surat berharga sebagai bentuk pengikat nasabah. Jaminan atau agunan adalah *second way out* dari nasabah atas pembiayaannya di bank sebagai mitigasi akhir jika kegiatan usaha nasabah tidak berjalan dengan baik yang mengakibatkan menurunnya pendapatan yang

berdampak pada tidak mampunya nasabah mengembalikan kewajiban kepada bank.

Bank ABC memberikan berbagai macam layanan. Salah satu layanan yang diberikan oleh Bank ABC adalah layanan agunan dan unit yang mengelola kegiatan agunan tersebut adalah Manajemen Risiko Bisnis. Saat ini kegiatan agunan yang dikelola oleh Manajemen Risiko Bisnis seperti pencatatan agunan nasabah sampai perhitungan nilai taksasi atau tafsiran masih dilakukan secara konvensional menggunakan excel oleh analis dan *supervisor*. Agunan merupakan salah satu faktor yang diperhitungkan dalam menilai aset atau properti yang diberikan oleh nasabah sebagai jaminan terhadap pembiayaan yang diajukan[3]. Data agunan yang akan di-*input* oleh analis seperti data agunan tanah, data agunan bangunan, foto lampiran dan foto lokasi agunan dilakukan dengan cara terkomputerisasi tetapi menggunakan excel dan hanya dapat dikerjakan di kantor saja. Setelah proses pendataan nasabah selesai, maka analis akan mencetak dan menyerahkan data nasabah tersebut kepada *supervisor* untuk dicek serta dikoreksi apabila terdapat kekeliruan. Jika terdapat kekeliruan, maka *supervisor* akan menyerahkan data kembali kepada analis untuk melakukan koreksi data. Kekeliruan atau masalah tersebut dapat dikarenakan adanya faktor *human error*, di mana analis harus melakukan *input* data dalam jumlah yang besar serta pencarian yang tergolong sulit di excel, dikarenakan tidak tersimpannya agunan nasabah yang telah melalui *approval* maupun *rejection*. Selain itu, masalah juga ditimbulkan karena kurang efektifnya penggunaan excel yang hanya boleh diakses di kawasan kantor saja. Lebih lanjut, adanya faktor inefisiensi waktu dikala proses *approval* berlangsung. Lamanya waktu yang dibutuhkan tentunya menghambat kinerja kerja di Bank ABC. Pada akhirnya, perbaikan atas kesalahan maupun kekeliruan akan dikembalikan kembali pada analis melalui proses yang sama hingga data dinilai sesuai oleh *supervisor*. Tetapi jika tidak terdapat kekeliruan, maka *supervisor* akan melakukan *approval* yang nantinya akan diteruskan kembali ke analis untuk dicetak dan ditandatangani oleh *supervisor*.

Untuk alasan ini, diperlukan suatu sistem yang dapat meningkatkan efisiensi kinerja Manajemen Risiko Bisnis dalam meningkatkan efektivitas pekerjaan, mempercepat proses pencatatan agunan, serta mengoptimalkan waktu. Dalam penelitian ini, akan dibuat suatu aplikasi berbasis web yang sesuai dengan permintaan pihak Bank ABC yang semua aplikasinya *web base* guna *access mobility* yang mudah dan dikembangkan menggunakan bahasa pemrograman PHP yang memiliki lisensi sumber terbuka (*open source*) dan *framework laravel* yang memiliki perkembangan pesat, struktur yang jelas serta memiliki dokumentasi yang lengkap untuk membantu dalam

proses pembangunan sistem informasi dengan model pengembangan sistem *Waterfall*[4] yang menekankan pengerjaan sistem yang sistematis dengan data agunan yang telah ditentukan jelas, proses yang dimiliki *laravel*, tidak menuntut perubahan dalam waktu cepat misalnya pada bagian fitur, serta desain yang telah ditetapkan, sehingga tidak membutuhkan banyak komunikasi dengan pihak Manajemen Risiko Bisnis sesuai dengan prinsip dari metode pengembangan *Waterfall*.

1.2. Rumusan Masalah

Dengan merujuk pada penjelasan sebelumnya, masalah yang diidentifikasi dalam tugas akhir ini adalah bagaimana merancang serta membangun Sistem Informasi Pengajuan Taksasi Agunan Manajemen Risiko Bisnis Pada Bank ABC yang yang digunakan oleh analis untuk melakukan pencatatan agunan nasabah menggunakan metode pengembangan *Waterfall*?

1.3. Tujuan

Tujuan dalam penelitian ini adalah merancang dan mengimplementasi Sistem Informasi Pengajuan Taksasi Agunan Manajemen Risiko Bisnis yang dapat memudahkan dalam pencatatan agunan nasabah menggunakan metode *Waterfall*.

2. TINJAUAN PUSTAKA

2.1. Dasar Teori

2.1.1. Taksasi

Nilai merujuk pada estimasi harga suatu barang atau benda. Taksasi juga dikenal sebagai estimasi, yaitu menentukan harga atau jumlah dengan perkiraan. Nilai juga dapat diinterpretasikan sebagai makna atau signifikansi dari suatu barang atau benda. Ini berarti bahwa barang atau benda akan memiliki nilai bagi seseorang jika memiliki makna bagi individu tersebut. Menurut Rambat Lupiyoadi, penilaian yang tinggi dapat mempengaruhi keputusan nasabah untuk memanfaatkan layanan pegadaian. Nasabah akan memberikan respon positif apabila nilai yang ditawarkan oleh produk atau layanan tersebut dapat memenuhi kebutuhan mereka[10].

2.1.2. Agunan

Agunan dalam konteks hukum perbankan didefinisikan dalam Pasal 1 angka 23 UU No. 10 Tahun 1998 tentang Perbankan sebagai jaminan tambahan yang diserahkan oleh Nasabah Debitur kepada Bank (Kreditur) dalam rangka mendapatkan fasilitas kredit atau pembiayaan berdasarkan Prinsip Syariah. Pasal 1 Angka 26 UU No. 21 Tahun 2008 tentang Perbankan Syariah menjelaskan bahwa Agunan merupakan jaminan tambahan, berupa benda bergerak atau benda tidak bergerak yang diberikan oleh pemilik Agunan

kepada Bank Syariah dan/atau UUS untuk menjamin pembayaran kewajiban Nasabah Penerima Fasilitas[11].

2.1.3. PHP

PHP merupakan singkatan dari Hypertext Preprocessor, sebuah bahasa pemrograman open source yang dikhususkan untuk pengembangan web dan dapat disisipkan dalam dokumen HTML.

PHP berperan sebagai bahasa skrip sisi server, yang berarti pemrosesan data dilakukan di sisi server. Dengan kata lain, server akan melakukan interpretasi terhadap skrip program dan mengirimkan hasilnya kepada klien yang mengajukan permintaan[13].

2.1.4. Apache

Apache merupakan sebuah perangkat lunak server web yang berfungsi dalam menangani permintaan dan *respons* HTTP, serta menyimpan informasi secara detail. Selain itu, *Apache* juga dikenal sebagai server web yang sederhana, terstruktur, mengikuti standar protokol HTTP, dan memiliki popularitas yang tinggi di kalangan pengguna. Fungsi utama *Apache* adalah untuk menghasilkan halaman web sesuai dengan yang dibuat oleh seorang *web programmer* menggunakan kode PHP.

2.1.5. Basis Data

Basis Data atau *Database* dianalogikan sebagai suatu tempat penyimpanan arsip. Istilah "basis" mengacu pada tempat di mana sesuatu berkumpul, bermarkas, atau bertempat tinggal, sementara "data" merujuk pada representasi fakta-fakta dalam dunia nyata yang menggambarkan suatu objek, seperti manusia (karyawan, siswa, mahasiswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, dan kondisi. Fakta-fakta ini direkam dalam berbagai bentuk, seperti angka, huruf, simbol, teks, gambar, suara, atau kombinasi di antaranya..

2.1.6. XMPP

XAMPP adalah sebuah perangkat lunak gratis yang kompatibel dengan berbagai sistem operasi, yang terdiri dari kumpulan program-program. Peran utama XAMPP adalah sebagai server lokal (*localhost*) yang melibatkan *Apache HTTP Server*, *MySQL Database*, dan juga interpreter bahasa seperti PHP dan Perl.

2.1.7. Unified Model Language (UML)

Unified Modelling Language (UML) merujuk pada sebuah bahasa standar yang digunakan di industri untuk memvisualisasikan, merancang model, dan mendokumentasikan sistem perangkat lunak. UML memiliki definisi notasi dan sintaksis yang digunakan. Notasi UML terdiri dari sejumlah bentuk khusus yang digunakan untuk menggambarkan berbagai diagram model perangkat lunak. Setiap bentuk memiliki makna khusus, dan sintaksis UML menentukan cara menggabungkan atau mengkombinasikan bentuk-bentuk tersebut. Pada dasarnya, notasi UML merupakan

pengembangan dari tiga notasi sebelumnya, yaitu Ivar Jacobson OOSE (*Object-Oriented Software Engineering*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Grady Booch OOD (*Object-Oriented Design*).

2.1.8. Class Diagram

Class merupakan suatu spesifikasi yang dapat dijadikan objek yang menjadi inti dalam pengembangan dan desain berbasis objek. *Class* menggambarkan keadaan sistem (atribut/properti) dan memiliki kemampuan untuk memanipulasi keadaan tersebut (metode/fungsi). Diagram class menggambarkan deskripsi dan struktur dari class, package, serta objek yang saling terkait.

2.1.9. Use Case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas suatu sistem dengan penekanan pada "apa" yang dilakukan oleh sistem, bukan "bagaimana" melakukannya. Aktor dengan sistem direpresentasikan dalam sebuah *use case*. Aktivitas yang biasanya ada pada *use case* seperti login ke sistem, membuat daftar belanja, dan lain sebagainya. Aktor dalam sistem dapat berupa manusia atau mesin yang berinteraksi dengan sistem lain untuk melaksanakan tugas-tugas tertentu. *Use case diagram* membantu dalam pengembangan persyaratan sistem, mengkomunikasikan desain yang diinginkan oleh klien, dan merancang *test case* untuk fitur-fitur yang ada dalam sistem.

2.1.10. Activity Diagram

Activity diagram berfungsi untuk mengilustrasikan berbagai aliran aktivitas dalam sistem yang sedang direncanakan, termasuk bagaimana aliran tersebut dimulai, keputusan yang mungkin terjadi, dan bagaimana aliran tersebut berakhir.

2.1.11. Sequence Diagram

Sequence Diagram menggambarkan urutan langkah-langkah proses dari setiap *use case* yang telah dibuat sebelumnya. Dalam diagram ini, juga terlihat pesan atau perintah yang dikirimkan beserta waktu eksekusinya.

2.1.12. Entity Relationship Diagram (ERD)

Menurut Simarmata (2010:67), *Entity Relationship Diagram* (ERD) merupakan alat utama dalam pemodelan data yang membantu mengorganisasi data dalam suatu proyek menjadi entitas-entitas dan menentukan hubungan antara entitas tersebut..

2.1.13. MySQL

MySQL adalah sebuah *Database Management System* (DBMS) yang menggunakan perintah berbasis perintah berbasis *Structured Query Language* (SQL). MySQL sering dipilih karena kemudahan dalam analisis. Keuntungan yang ditawarkan mencakup

kecepatan, kehandalan, serta dukungan yang luas dari pengembang perangkat lunak dan aplikasi di *platform* web yang menggunakan *database open source* ini.

2.1.14. *Laravel*

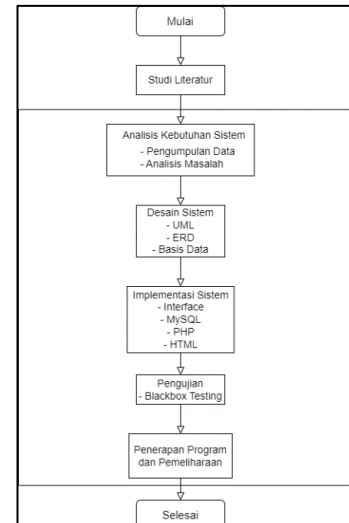
Laravel merupakan suatu kerangka kerja yang menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan digunakan untuk mengembangkan aplikasi web. *Framework* ini mengadopsi konsep *Model View Controller* (MVC) yang efisien dan mudah dipelajari (Bandirmali, Necla, 2018). Tujuan dari pengembangan *Laravel* adalah untuk meningkatkan kualitas aplikasi dengan mengurangi biaya pengembangan, mempermudah pemeliharaan, dan meningkatkan produktivitas melalui penggunaan kode program yang terstruktur dan rapi [14].

2.1.15. Metode Pengembangan *Waterfall*

Definisi agunan dalam konteks hukum perbankan dapat ditemukan dalam Pasal 1 angka 23 UU No. 10 Tahun 1998 tentang Perbankan, yang menjelaskan bahwa agunan adalah jaminan tambahan yang diberikan oleh Nasabah Debitur kepada Bank (Kreditur) dalam rangka mendapatkan fasilitas kredit atau pembiayaan berdasarkan Prinsip Syariah. Di sisi lain, Pasal 1 Angka 26 UU No. 21 Tahun 2008 tentang Perbankan Syariah menjelaskan bahwa agunan merupakan jaminan tambahan berupa benda bergerak atau benda tidak bergerak yang diserahkan oleh pemilik agunan kepada Bank Syariah dan/atau UUS (Usaha Unit Syariah) sebagai bentuk jaminan untuk menjamin pelunasan kewajiban Nasabah Penerima Fasilitas [11].

3. METODE PENELITIAN

Dalam melakukan penelitian, diperlukan suatu metode yang dapat digunakan sebagai acuan atau pedoman untuk pengembangan sistem. Beberapa metode yang umum digunakan adalah *Waterfall*, *SDLC*, *Prototyping*, *Agile* dan lain sebagainya. Namun, penelitian tugas akhir ini menggunakan pengembangan *Waterfall* yang tahapannya diawali dengan analisis kebutuhan sistem yaitu proses pengumpulan data dilakukan melalui wawancara, observasi, dan studi pustaka serta proses analisis masalah menggunakan *fishbone analysis*. Kemudian dilanjutkan tahap desain sistem, implementasi sistem, pengujian, serta pemeliharaan sistem. Berikut merupakan Gambar 3.1 Diagram Alir Penelitian dengan metodologi pengembangan model *Waterfall* yang memiliki 5 tahapan:



Gambar 1. Diagram alir penelitian

3.1. Studi Literatur

Langkah awal dalam penelitian tugas akhir ini adalah studi literatur yang berfungsi untuk mencari dan mengumpulkan keterangan dan informasi melalui buku, jurnal, skripsi, internet, serta studi-studi sebelumnya yang terkait dengan pengembangan sistem informasi dengan model *Waterfall* dan *framework laravel*. *Output* dari studi literatur ini berupa terkumpulnya referensi yang relevan berdasarkan perumusan masalah serta tujuan dan sebagai landasan teori dalam menjalankan penelitian.

3.2. Analisis Kebutuhan Sistem

Tahap awal dari penelitian tugas akhir ini adalah menganalisis kebutuhan sistem. Perlunya menganalisis apa saja kebutuhan sistem berguna untuk memudahkan peneliti melakukan proses pengembangan sistem yang dibangun yaitu Rancang Bangun Sistem Informasi Pengajuan Taksasi Agunan Manajemen Risiko Bisnis Pada Bank ABC. Kemudian untuk data yang dibutuhkan pada sistem hanya data agunan nasabah berupa dokumen *hardcopy* dan *softcopy*. Sistem ini bertujuan untuk memudahkan admin pada Manajemen Risiko Bisnis dalam mengelola data agunan nasabah dan menyediakan *history* agunan nasabah. Ada 3 tahap pengumpulan kebutuhan dalam penelitian tugas akhir ini yakni sebagai berikut:

1. Wawancara
2. Observasi
3. Studi Pustaka

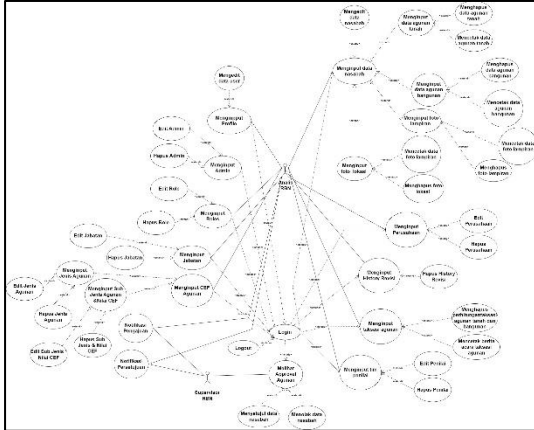
3.3. Desain Sistem

Setelah tahap perencanaan selesai dan didapat hasil yang diperlukan untuk melakukan pengembangan sistem, tahap selanjutnya adalah pembuatan desain sistem guna penentuan kebutuhan fungsionalnya.

Penelitian tugas akhir ini dirancang menggunakan *Unified Model Language (UML)*.

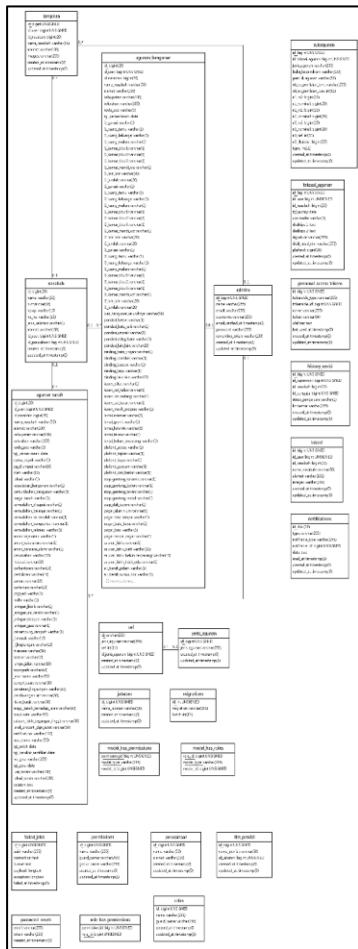
3.3.1. Use Case Diagram

Dalam *use case diagram* yang diberikan dapat dilihat masing-masing perilaku dari analis sistem dan *supervisor*. *Use case diagram* rancang bangun sistem informasi taksasi agunan Manajemen Risiko Bisnis terlihat pada ilustrasi yang disajikan yaitu Gambar 2:



Gambar 2. Use Case Diagram

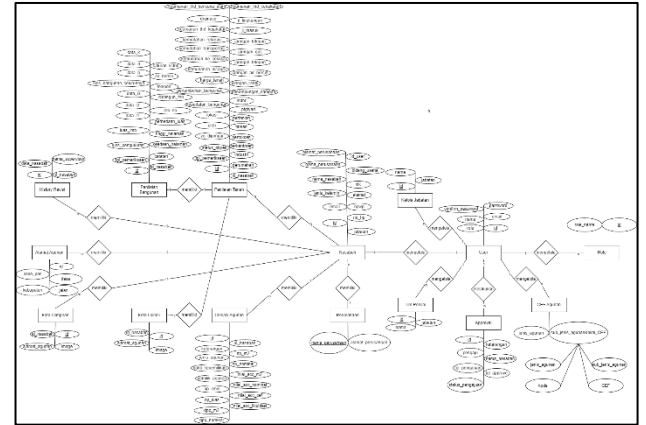
3.3.2. Class Diagram



Gambar 3. Class Diagram

Ilustrasi Gambar 3 di atas merupakan *Class diagram* yang berfungsi untuk mendefinisikan *class-class* yang akan dibuat dalam merancang sistem informasi taksasi agunan Manajemen Risiko Bisnis sebagai penggambaran struktur sistem dengan menggunakan model arsitektur *Model-View-Controller (MVC)*.

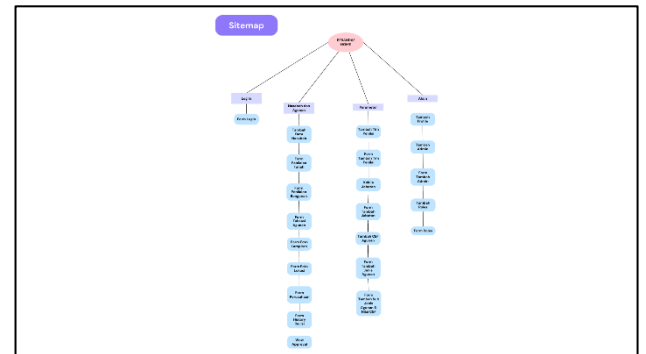
3.3.3. Entity Relationship Diagram



Gambar 4. Entity Relationship Diagram (ERD)

Desain struktur *Entity Relationship Diagram (ERD)* dari Sistem Informasi Pengajuan Taksasi Agunan Manajemen Risiko Bisnis Pada Bank ABC dapat dilihat pada ilustrasi Gambar 4 di atas.

3.3.4. Sitemap Sistem



Gambar 5. Sitemap Sistem

Pada Gambar 5 di atas merupakan sitemap dari sistem informasi taksasi agunan Manajemen Risiko Bisnis yang memiliki beberapa fitur yaitu fitur Taksasi Agunan, Parameter, dan Pengaturan Akun. Dalam ketiga fitur tersebut terdapat beberapa aksi yang dapat dilakukan oleh analis seperti pada gambar. Untuk dapat mengakses fitur maupun aksi yang tersedia dalam sistem ini, diperlukan proses login terlebih dahulu guna menjamin keamanan sistem sehingga sistem dapat melakukan verifikasi pengguna.

3.4. Implementasi Sistem

Setelah tahap desain selesai, langkah berikutnya adalah tahap implementasi sistem. Tahap ini melibatkan penggunaan bahasa pemrograman skrip seperti PHP untuk membangun web, serta MySQL sebagai basis data yang menggunakan perintah SQL (*Structured*

Query Language) untuk pengelolaan data dalam Database, kemudian kerangka kerja atau framework yang digunakan adalah Laravel guna pembangunan aplikasi yang dinamis.

3.5. Pengujian Black Box Testing

Tahap pengujian sistem dimaksudkan untuk melakukan pengujian terhadap sistem yang telah selesai dibangun pada tahap implementasi. Pada metode pengujian sistem, digunakan pendekatan *Black Box Testing* atau yang sering disebut sebagai *functional testing*. Pengujian ini bertujuan untuk mengevaluasi apakah fungsi-fungsi sistem telah berjalan dengan baik atau tidak yang nantinya akan merancang *test case* berdasarkan informasi yang ada. Pada pengujian ini melibatkan pihak Manajemen Risiko Bisnis selaku *client* yaitu analis dan *supervisor* yang berjumlah kurang lebih 4 orang untuk melakukan pengujian sistem.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Database

Dalam tahap implementasi database sistem informasi taksasi agunan Manajemen Risiko Bisnis terbentuk 24 tabel yang mana 3 di antaranya merupakan tabel bawaan framework Laravel itu sendiri. Gambar 6 berikut menunjukkan database sistem.

Table	Action
admins	Browse Structure Search Insert Empty Drop
agunan_bangunan	Browse Structure Search Insert Empty Drop
agunan_tanah	Browse Structure Search Insert Empty Drop
cef	Browse Structure Search Insert Empty Drop
failed_jobs	Browse Structure Search Insert Empty Drop
history_revisi	Browse Structure Search Insert Empty Drop
jabatan	Browse Structure Search Insert Empty Drop
jenis_agunan	Browse Structure Search Insert Empty Drop
lampiran	Browse Structure Search Insert Empty Drop
lokasi	Browse Structure Search Insert Empty Drop
migrations	Browse Structure Search Insert Empty Drop
model_has_permissions	Browse Structure Search Insert Empty Drop
model_has_roles	Browse Structure Search Insert Empty Drop
nasabah	Browse Structure Search Insert Empty Drop
notifications	Browse Structure Search Insert Empty Drop
password_resets	Browse Structure Search Insert Empty Drop
permissions	Browse Structure Search Insert Empty Drop
personal_access_tokens	Browse Structure Search Insert Empty Drop
perusahaan	Browse Structure Search Insert Empty Drop
roles	Browse Structure Search Insert Empty Drop
role_has_permissions	Browse Structure Search Insert Empty Drop
subagunan	Browse Structure Search Insert Empty Drop
taksasi_agunan	Browse Structure Search Insert Empty Drop
tim_penilai	Browse Structure Search Insert Empty Drop
24 tables	Sum

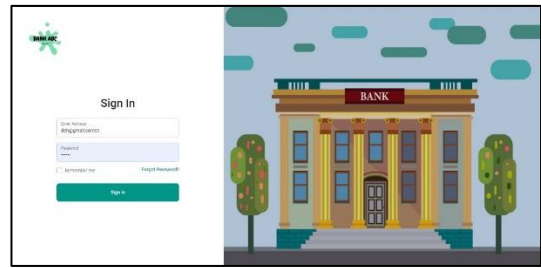
Gambar 6. Implementasi Database

4.2. Implementasi Class

Implementasi *class* dirancang berdasarkan class diagram yang dibentuk. Class yang terbentuk ke dalam tiga bagian yaitu, *class controller* yang berfungsi untuk memberikan perintah, *class model* yang akan mengolah dan mencari data dalam database, dan *class view* yang akan mengurus tampilan sistem.

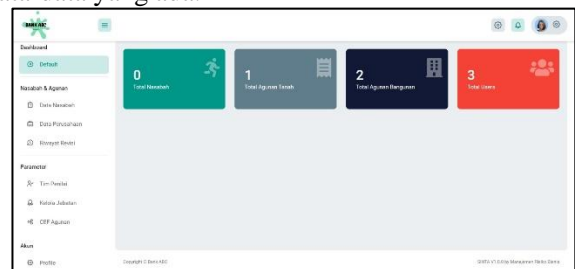
4.3. Implementasi Interface

Berikut adalah implementasi *interface* sistem yang telah dikembangkan:



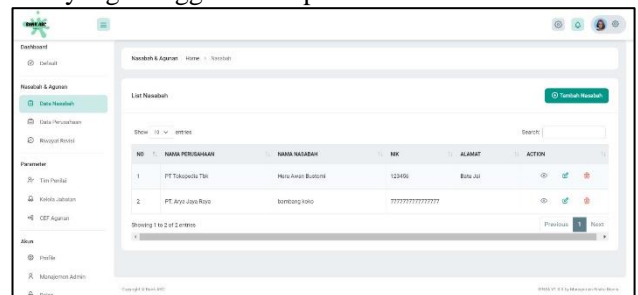
Gambar 7. Halaman Login

Ilustrasi Gambar 7 di atas yaitu halaman *login* dari *user*. Di halaman ini terdapat *role* yang membedakan *user* dengan cara memasukkan *username* atau email beserta *password* yang telah ter-list agar terjaminnya data-data yang ada.



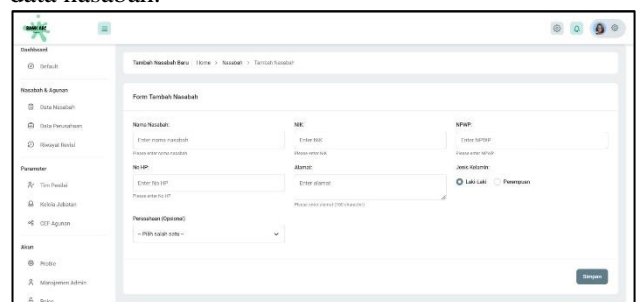
Gambar 8. Halaman Home

Ilustrasi Gambar 8 di atas yaitu halaman *home*. Di halaman ini terdapat informasi tentang jumlah total nasabah, total agunan tanah dan bangunan, serta total *user* yang menggunakan aplikasi ini.



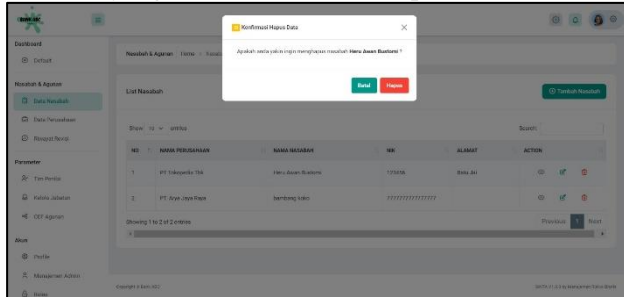
Gambar 9. List Nasabah

Ilustrasi Gambar 9 di atas yaitu halaman *list* nasabah. Di halaman ini terdapat informasi mengenai nasabah dan juga beberapa aksi seperti detail nasabah, tambah dan edit nasabah yang menampilkan *interface* yang sama, dan terakhir adalah aksi untuk menghapus data nasabah.



Gambar 10. Tambah Nasabah

Ilustrasi Gambar 10 di atas yaitu halaman tambah nasabah. Di halaman ini tersedia sebuah *form* yang harus diisi analis guna menambahkan data nasabah. Pada saat akan melakukan edit data nasabah, maka *interface* yang muncul akan sama seperti ini.



Gambar 11. Hapus Nasabah

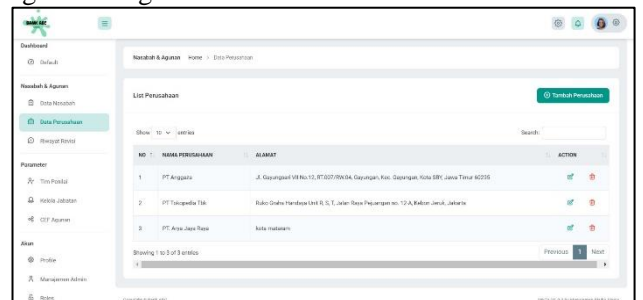
Ilustrasi Gambar 11 di atas yaitu halaman hapus nasabah. Pada halaman ini analis dapat menghapus data nasabah dengan cara memilih menu aksi dan data nasabah yang telah dihapus tidak akan muncul pada halaman *list* nasabah lagi.

Gambar 12. *Input* Tanah

Ilustrasi Gambar 12 di atas yaitu halaman *input* agunan tanah. Pada halaman ini terdapat *form* mengenai taksasi agunan tanah yang harus dilengkapi analis apabila ingin menambah data taksasi agunan tanah nasabah.

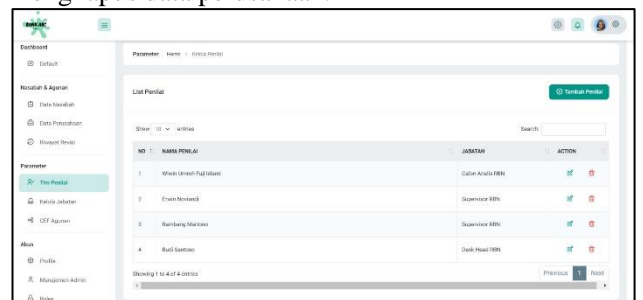
Gambar 13. *Input* Bangunan

Ilustrasi Gambar 13 di atas merupakan halaman *input* agunan bangunan. Pada halaman ini terdapat *form* mengenai taksasi agunan bangunan yang harus dilengkapi analis apabila ingin menambah data taksasi agunan bangunan nasabah.



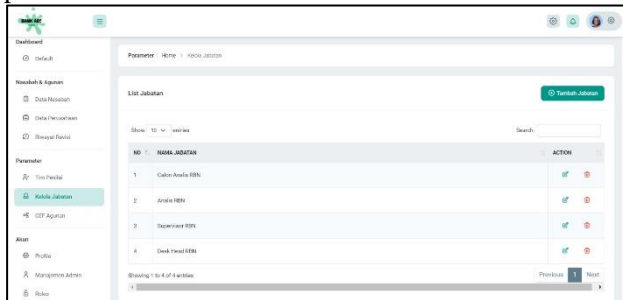
Gambar 14. *List* Perusahaan

Ilustrasi Gambar 14 di atas merupakan halaman *list* perusahaan. Pada halaman ini terdapat informasi mengenai nama perusahaan dan juga beberapa aksi seperti tambah dan edit nama perusahaan yang menampilkan *interface* yang sama seperti apabila ingin menambah nasabah *dengan form* yang disesuaikan dengan jenis *list*, dan terakhir adalah aksi untuk menghapus data perusahaan.



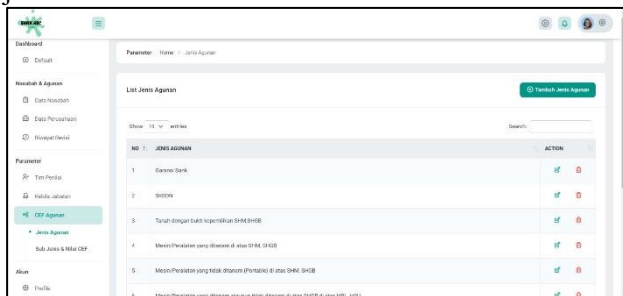
Gambar 15. *List* Penilai

Ilustrasi Gambar 15 di atas merupakan halaman *list* penilai. Pada halaman ini terdapat informasi mengenai nama penilai dan juga beberapa aksi seperti tambah dan edit nama perusahaan yang menampilkan *interface* yang sama seperti apabila ingin menambah nasabah dengan *form* yang disesuaikan dengan jenis *list*, dan terakhir adalah aksi untuk menghapus data penilai.



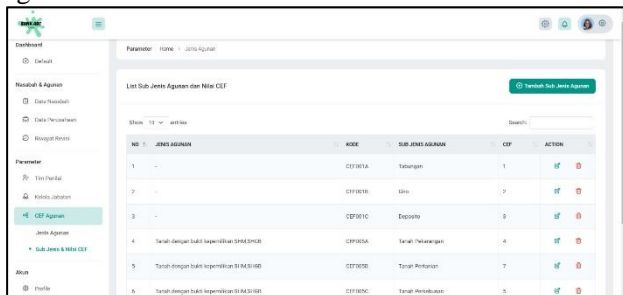
Gambar 16. *List* Jabatan

Ilustrasi Gambar 16 di atas merupakan halaman *list* jabatan. Pada halaman ini terdapat informasi mengenai nama jabatan dan juga beberapa aksi seperti tambah dan edit nama jabatan yang menampilkan *interface* yang sama seperti apabila ingin menambah nasabah dengan *form* yang disesuaikan dengan jenis *list*, dan terakhir adalah aksi untuk menghapus data jabatan.



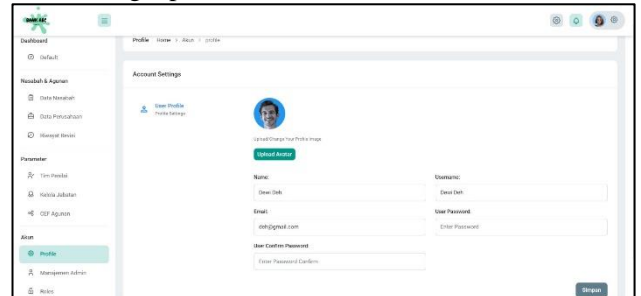
Gambar 17. *List* Jenis Agunan

Ilustrasi Gambar 17 di atas merupakan halaman *list* jenis agunan. Pada halaman ini terdapat informasi mengenai jenis agunan dan juga beberapa aksi seperti tambah dan edit jenis agunan yang menampilkan *interface* yang sama seperti apabila ingin menambah nasabah dengan *form* yang disesuaikan dengan jenis *list*, dan terakhir adalah aksi untuk menghapus data jenis agunan.



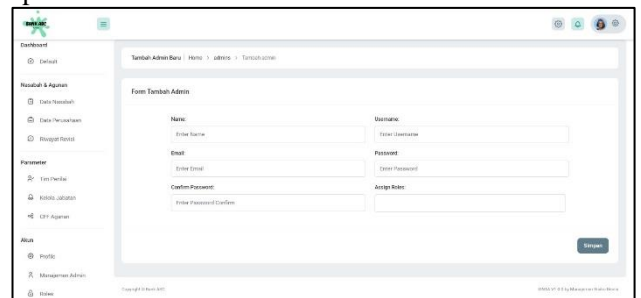
Gambar 18. *List* CEF

Gambar 18 di atas merupakan halaman *list* CEF. Pada halaman ini terdapat informasi mengenai jenis agunan dan juga beberapa aksi seperti tambah dan edit CEF yang menampilkan *interface* yang sama seperti apabila ingin menambah nasabah dengan *form* yang disesuaikan dengan jenis *list*, dan terakhir adalah aksi untuk menghapus data CEF.



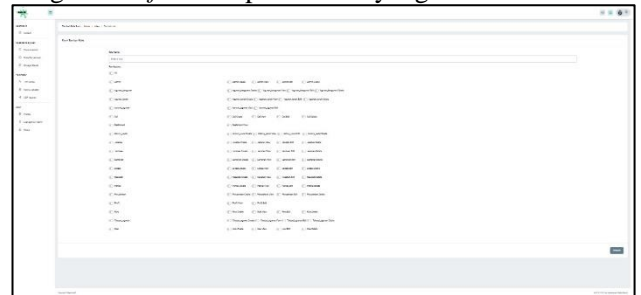
Gambar 19. *Profile*

Ilustrasi Gambar 19 di atas merupakan halaman *profile*. Pada halaman ini terdapat *form* yang harus dilengkapi analisis apabila ingin menambah *user* pada aplikasi.



Gambar 20. Tambah Admin

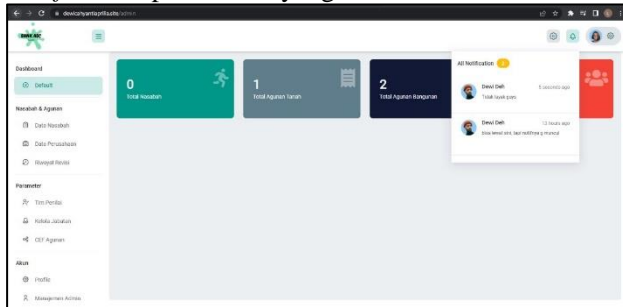
Ilustrasi Gambar 20 di atas yaitu halaman yang digunakan untuk menambah admin. Di halaman ini tersedia *form* yang harus dilengkapi analisis untuk menambah admin pada aplikasi. Pada saat akan melakukan edit data admin, maka *interface* yang muncul akan sama seperti ini. Kemudian untuk *list* admin sendiri, halaman *interface*-nya akan mirip dengan *interface list* pada menu yang lain.



Gambar 21. Tambah *Role*

Ilustrasi Gambar 21 di atas merupakan halaman tambah *role*. Pada halaman ini terdapat *form* yang harus dilengkapi analisis dan terdapat beberapa pilihan *roles* yang dapat digunakan oleh analisis untuk menentukan tindakan apa saja untuk *user* lain pada aplikasi ini. Pada saat akan melakukan edit data *role*, maka *interface*

yang muncul akan sama seperti ini. Kemudian untuk *list role* sendiri, halaman *interface*-nya akan mirip dengan *interface list* pada menu yang lain.



Gambar 22. Notifikasi

Ilustrasi Gambar 22 di atas merupakan halaman notifikasi. Pada halaman ini ditampilkan notifikasi yang berasal dari *user*. Notifikasi akan terkirim kepada *supervisor* ketika analis menambah *history* revisi dan *supervisor* akan menindaklanjutinya. Kemudian setelah *supervisor* melakukan *approval* atau *rejection*, *supervisor* akan mengirim data tersebut dengan menekan *button* kirim. *Interface* dari halaman notifikasi analis dan *supervisor* menampilkan informasi yang sama.

4.4. Black Box Testing

Black box testing adalah metode pengujian yang melibatkan pengujian fungsionalitas dari fitur-fitur yang ada dalam sistem setelah tahap implementasi dilakukan. Dari setiap fitur yang telah dikelompokkan ke dalam 9 kategori fitur secara umum dan kemudian diuji, semuanya menghasilkan kesimpulan “Valid” yang berarti setiap fiturnya tidak ditemukan *error* dan telah dapat berjalan dengan baik. Hasil pengujian dengan menggunakan metode *black box testing* adalah sebagai berikut:

Tabel 1. Hasil Pengujian *Black Box Testing*

No	Fitur	Kesimpulan
1	Fungsi Login	Valid
2	Fungsi Data Nasabah	Valid
3	Fungsi Data Perusahaan	Valid
4	Fungsi Tim Penilai	Valid
5	Fungsi Kelola Jabatan	Valid
6	Fungsi CEF	Valid
7	Fungsi Profile	Valid
8	Fungsi Manajemen Admin	Valid
9	Fungsi Role	Valid

4.5. Mean Opinion Scores (MOS)

MOS merupakan testing yang dilakukan dari sisi *client*. Pada tahap ini, akan dilakukan pengujian terhadap kualitas dan performa sistem yang telah dikembangkan. Hasil akhir dari sistem ini akan digunakan sebagai pertimbangan apakah sistem layak untuk digunakan atau tidak.

Terdapat 4 orang responden yang akan menjawab pertanyaan setelah melakukan uji coba aplikasi. Berikut merupakan daftar pertanyaan yang diajukan sebanyak 13 pertanyaan.

No	Pertanyaan	Kategori	SB	B	C	KB	TB	Jumlah	Rata-rata
1	Sehuruh fungsi atau fitur Sistem Informasi Takasi Agunan berjalan dengan baik	Functionality	20	0	0	0	0	20	5
2	Fitur yang ada dalam aplikasi sesuai dengan requirement awal	Functionality	20	0	0	0	0	20	5
3	Sistem mudah diakses oleh user	Functionality	20	0	0	0	0	20	5
4	Tidak dihasilkan error dan bugs berarti oleh sistem	Reliability	15	4	0	0	0	19	4,75
5	User merasa nyaman ketika menggunakan sistem	Reliability	10	8	0	0	0	18	4,5
6	Sistem memiliki tampilan serta fungsi yang mudah dipahami	Reliability	15	4	0	0	0	19	4,75
7	Pengoperasian sistem mudah	Reliability	20	0	0	0	0	20	5
8	Letak tombol dari fitur yang ada pada sistem mudah untuk diingat	Reliability	10	8	0	0	0	18	4,5
9	Sistem dapat diakses dan bekerja melalui semua jaringan	Efficiency	20	0	0	0	0	20	5
10	Fitur pada sistem dapat berjalan efektif dan efisien (wifi/data cellular)	Efficiency	20	0	0	0	0	20	5
11	Data yang ada dalam sistem terjamin keamanannya	Portability	5	12	0	0	0	17	4,25
12	Sistem dapat melakukan komputasi atau merespon query berupa perintah dengan cepat	Portability	15	0	3	0	0	18	4,5
13	Tampilan sistem tetap teratur saat diakses pada berbagai platform	Portability	20	0	0	0	0	20	5

Gambar 23. Daftar Pertanyaan

Jumlah dari setiap pertanyaan dihitung berdasarkan pilihan kepuasan client dikalikan dengan bobot setiap variable dari 1 sampai 5 sesuai dengan Gambar 24 berikut.

MOS	Kategori	Bobot
SB / SS	Sangat Baik/Sangat Setuju	5
B / S	Baik/Setuju	4
C	Cukup	3
KB / KS	Kurang Baik/ Kurang Setuju	2
TB / TS	Tidak Baik/Tidak Setuju	1

Gambar 24. Bobot Penilaian MOS

Setelah pengujian selesai dilakukan dan didapatkan bobot rata-rata dari setiap pertanyaan, maka akan dilakukan perhitungan persentase setiap pertanyaan menggunakan metode MOS seperti pada Gambar 25 berikut.

Pertanyaan	Perhitungan	Persentase
P1.	$\frac{5}{5} 100\%$	100%
P2.	$\frac{5}{5} 100\%$	100%
P3.	$\frac{5}{5} 100\%$	100%
P4.	$\frac{4,75}{5} 100\%$	95%
P5.	$\frac{4,5}{5} 100\%$	90%
P6.	$\frac{4,75}{5} 100\%$	95%
P7.	$\frac{5}{5} 100\%$	100%
P8.	$\frac{4,5}{5} 100\%$	90%
P9.	$\frac{5}{5} 100\%$	100%
P10.	$\frac{5}{5} 100\%$	100%
P11.	$\frac{4,25}{5} 100\%$	84%
P12.	$\frac{4,5}{5} 100\%$	90%
P13.	$\frac{5}{5} 100\%$	100%
Rata-rata		95,69%

Gambar 25. Hasil Persentase Pertanyaan

Hasil yang didapat setelah dilakukannya perhitungan menunjukkan bahwa rata-rata persentase yang didapat sistem sebesar 95,69%. Berdasarkan interpretasi skor tersebut, maka digolongkan sangat baik dan sistem layak digunakan.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan analisis pembahasan terhadap penelitian yang telah dilakukan, pada bab dihasilkan beberapa kesimpulan sebagai berikut:

1. Pengembangan Sistem Informasi Taksasi Agunan Manajemen Risiko Bisnis Pada Bank ABC menggunakan metode *Waterfall* cocok dan selesai tepat dengan jadwal yang telah ditentukan. Pembangunan sistem yang ditunjang dengan *framework Laravel* disertai dengan data, tidak menghambat proses pengembangan walaupun alur metode yang digunakan linear. Sehingga hasil akhir dari sistem telah sesuai dengan *requirement* awal yang diminta.
2. Fitur-fitur yang mendukung dan memudahkan pengguna menggunakan sistem ini seperti fitur untuk pencatatan agunan nasabah, kelola tim penilai, jabatan dan CEF agunan, kemudian kelola akun baik itu *profile*, manajemen admin, dan *roles* telah memenuhi kebutuhan pengguna sehingga dapat digunakan sebaik mungkin.
3. Berdasarkan penelitian yang telah dilakukan, aplikasi ini memudahkan kinerja Manajemen

Risiko Bisnis pada Bank ABC untuk mengelola kegiatan taksasi yang dapat dibuktikan dengan hasil pengujian menggunakan *Black Box Testing* dan metode *Mean Opinion Scores (MOS)* kepada calon pengguna aplikasi yaitu Manajemen Risiko Bisnis bahwa semua fitur dan proses telah berjalan dengan baik serta layak untuk digunakan.

5.2. Saran

Adapun saran untuk sistem setelah dilakukan pengembangan yang dapat dijadikan perbaikan ke depannya adalah sebagai berikut:

1. Untuk pengembangan sistem kedepannya, diharapkan aplikasi dapat dikembangkan menjadi *cross platform* agar sistem dapat diimplementasi ke dalam *mobile app*.
2. Untuk pengembangan sistem dalam hal notifikasi, diharapkan kedepannya fitur ini dapat mengirimkan notifikasi melalui media seperti whatsapp dan email.

DAFTAR PUSTAKA

- [1] Nilai Agunan Terhadap Pemberian Kredit Bank Bri Unit Hasanuddin Parepare (Analisis Ekonomi Islam). 2018.
- [2] A. Muhammad Aris, "Jurnal Wanprestasi Dalam Pelaksanaan Perjanjian Konsinyasi Di Dapur Roti Bu Haryati," *Lincoln Arsyad*, vol. 3, no. 2, pp. 1–46, 2014, [Online]. Available: <http://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127>
- [3] Ansori, *Analisis Penilaian Agunan Pada Pembiayaan "Ib Kepemilikan" Di Pt. Bprs Gunung Slamet Cilacap Laporan*, vol. 3, no. April. 2015.
- [4] A. S. Kirsan and N. N. Arisa, "Perancangan Sistem Informasi Manajemen Badan Amal Menggunakan Metode *Waterfall*," vol. 16, no. 1, pp. 63–77, 2022.
- [5] A. Asroni, A. Setiawan, and B. D. Alfian, "Implementasi Framework *Laravel* Pada Sistem Informasi ZIS," *Pros. Semin. Nas. ...*, pp. 2270–2274, 2020, doi: 10.18196/ppm.311.349.
- [6] A. Haq, "Sistem Informasi Penggajian PT. Kalisha Utama Ghani Cilacap Menggunakan Framework *Laravel*," vol. 4, no. 2, pp. 153–162, 2020.
- [7] S. Yudhistira, "Implementasi Framework *Laravel* Untuk Membangun Sistem Administrasi Pembayaran Air Pada Ppab Perum Dolog Dengan Metode *Waterfall*," 2019.
- [8] I. C. Woei, "Penerapan *Laravel* Framework Dalam Perancangan Sistem Informasi Promosi Produk Unggulan UKM Berbasis Web (Studi Kasus Dinas Perindustrian Perdagangan dan UMKM Kota Salatiga)," no. June, 2016.

- [9] M. L. Sholihin, "Sistem Informasi Desa Berbasis Website (Studi Kasus Kelurahan Lalung)," pp. 1–22, 2021.
- [10] H. Maros And S. Juniar, "Pengaruh Nilai Taksasi Barang Agunan Terhadap Minat Nasabah Dalam Pembiayaan Murabahah (Studi Kasus PT. BPRS Al Washliyah)," pp. 1–23, 2016.
- [11] I. L. Fitriani, "Jaminan Dan Agunan Dalam Pembiayaan Bank Syariah Dan KreditBank Konvensional," *J. Huk. Pembang.*, vol. 47, no. 1, p. 134, 2017, doi: 10.21143/jhp.vol47.no1.138.
- [12] A. F. Sallaby and I. Kanedi, "Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter," *J. Media Infotama*, vol. 16, no. 1, pp. 48–53, 2020, doi: 10.37676/jmi.v16i1.1121.
- [13] I. Rahmat, "Sistem Informasi Perpustakaan Online Berbasis Web," *J. Ilm. Syi'ar*, vol. 18, no. 1, p. 23, 2018, doi: 10.29300/syr.v18i1.1568.
- [14] M. Z. Abdullah, M. Astiningrum, Y. Ariyanto, D. Puspitasari, and A. N. Asri, "Rancang Bangun Sistem Informasi Akuntansi Berbasis Website menggunakan Framework *Laravel*," *J. Sains, Teknol. dan Ind.*, vol. 18, no. 1, p. 49, Dec. 2020, doi: 10.24014/sitekin.v18i1.11313.
- [15] L. Kelen, "Implementasi Model-View-Controller (Mvc) Pada Ujian Online Melalui Penerapan Framework Codeigniter," *J. Pendidik. Teknol. Inf.*, vol. 1, no. 1, pp. 10–16, 2018, doi: 10.37792/jukanti.v1i1.5.