

MODEL PENGEMBANGAN APLIKASI PEMBAYARAN ANGSURAN PINJAMAN *ONLINE* MENGGUNAKAN PHP-MYSQL DENGAN METODE *OBJECT ORIENTED PROGRAMMING*

Whisnumurti Adhiwibowo¹, April Firman Daru²

^{1, dan 2} Universitas Semarang

Abstrak—Sistem pembayaran angsuran berbasis web pada suatu koperasi merupakan sistem yang bersifat dinamis dalam arti akan selalu berkembang dan kompleks. Metode Pemrograman prosedural memiliki kelemahan saat mengembangkan aplikasi yang kompleks. Ketika terjadi permasalahan, penanganannya menjadi sulit karena program terdiri dari banyak fungsi dan pada saat mengubah suatu fungsi akan mengubah fungsi fungsi yang lain. Tidak hanya saat terjadi permasalahan, suatu sistem berbasis pemrograman prosedural juga susah apabila akan dikembangkan. Oleh karena itu di pandang perlu untuk mengembangkan Sistem dengan metode object Oriented Programming (OOP), merupakan metode pemrograman berorientasi obyek. Sistem yang di bangun dengan metode ini terdiri dari banyak obyek yang saling berhubungan. Perancangan yang dipakai menggunakan Unified Modeling Language (UML), suatu metodologi untuk mengembangkan sistem dengan pendekatan OOP. Pengembangan Sistem ini menggunakan Hypertext Preprocessor (PHP). Metode pengembangan sistem yang digunakan adalah Waterfall. Hasil dari penelitian ini adalah terbentuknya sistem pembayaran transaksi berbasis web dengan metode OOP, sehingga dapat lebih mudah pemeliharaan dan pengembangannya.

Kata Kunci : Metode *OOP*, Model Pengembangan Aplikasi, Sistem Pembayaran Online.

Abstract—Web-based installment payment system in a cooperative is a dynamic system in the sense that will always grow and complex. Procedural Programming methods have weaknesses when developing complex applications. When problems occur, handling becomes difficult because the program consists of many functions and when changing a function will change the function of another function. Not only when there is a problem, a procedural programming based system is also difficult if it will be developed.

Therefore, in view of the need to develop the System with object Oriented Programming (OOP) method, is an object-oriented programming method. The system built by this method consists of many interconnected objects. The design used by Unified Modeling Language (UML), a methodology for developing a system with the OOP approach. Development This system uses Hypertext Preprocessor (PHP). System development method used is Waterfall. The purpose of this research is the establishment of web-based transaction payment system with OOP method, so it can be easier maintenance and development.

Keywords: *OOP Method, Application Development Model, Online Payment System*

Pendahuluan

1.1. Latar Belakang

Pemrograman prosedural hanya efektif bila digunakan untuk membangun sistem yang sederhana. Pengembangan sistem pembayaran angsuran online dengan menggunakan pemrograman prosedural memiliki kelemahan, diantaranya susah menemukan masalah pada program apabila suatu fungsi tidak aktif. Apabila fungsi yang tersedia diubah akan mempengaruhi fungsi fungsi yang lain. Selain itu proses pengembangan sistem sulit dikerjakan, padahal sebuah sistem akan selalu berkembang mengikuti kebutuhan yang semakin kompleks. Untuk menterjemahkan business models ke programming models juga cukup sulit.

Object Oriented Programming (OOP) merupakan pengembangan dari pemrograman prosedural, OOP merupakan model pemrograman menggunakan obyek. OOP menggunakan class dan obyek yang dapat digunakan berulang ulang, apabila ada pengembangan sebuah sistem tinggal membuat obyek baru, sehingga

ketika terjadi permasalahan lebih mudah untuk mengatasinya. OOP sangat efektif untuk mengembangkan sistem yang kompleks.

Sistem pembayaran angsuran akan selalu berkembang seiring dengan meningkatnya jumlah pinjaman dan jumlah peminjam. Dengan menggunakan OOP, Perubahan sistem dapat dilakukan oleh programer yang sama atau berbeda. Dewasa ini perkembangan teknologi internet semakin maju, sehingga sistem yang dikembangkan cenderung berbasis web, agar dapat diakses secara continue.

Berdasarkan latar belakang permasalahan diatas, maka penulis akan mengembangkan Sistem pembayaran angsuran online menggunakan model OOP dengan metode pengembangan sistem Waterfall. Sistem yang dikembangkan dengan OOP akan dapat dikembangkan tidak hanya programer yang bersangkutan, tapi programer yang berbeda. Hal ini karena model OOP berbasis obyek, fungsi fungsi yang ada dibagi dalam beberapa class, sehingga penanganannya menjadi lebih mudah, selain itu apabila

terjadi bug pada program, dapat lebih mudah memmemukan dan memperbaikinya.

Dengan adanya sistem pembayaran angsuran berbasis web menggunakan metode OOP, akan memudahkan proses transaksi, jika terjadi permasalahan pada fungsi sistem, lebih cepat menemukan dan memperbaiki bug. Selain itu sistem yang dikembangkan dengan metode OOP dapat dikembangkan dengan lebih mudah

1.2. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah terbentuknya sistem pembayaran angsuran pinjaman berbasis web dengan metode OOP, sehingga dapat lebih mudah pengembangannya apabila akan ditambahkan fungsi fungsi yang baru.

1.3. Manfaat

Penerapan suatu teori ke dalam aplikasi, khususnya teori tentang Pemrograman berorientasi obyek dalam pengembangan sistem informasi pembayaran online yang siap dikembangkan dengan penambahan fungsi fungsi baru seperti penjualan beserta laporan keuangannya.

I. KAJIAN PUSTAKA

2.1. Sistem Informasi

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah Satu kesatuan yang terdiri dari beberapa komponen atau elemen yang memiliki aliran hubungan untuk maksud dan tujuan yang sama. Istilah ini sering digunakan untuk menggambarkan suatu entitas yang saling terhubung antara satu dengan yang lainnya sehingga semuanya saling berinteraksi. Dalam ilmu komputer dan ilmu informasi, sistem adalah Software System yang terdiri dari komponen komponen terstruktur dan saling berkomunikasi.

Informasi (disingkat Info) adalah yang menginformasikan. Dengan kata lain, itu adalah jawaban untuk beberapa jenis pertanyaan. Hal ini juga berkaitan dengan dari mana data dan pengetahuan dapat diperoleh, data yang memiliki nilai nilai dan berkaitan dengan parameter, pengetahuan yang menandakan hal hal nyata atau abstrak [9]. Informasi adalah pesan yang memiliki arti berbeda dalam konteks yang berbeda [3]. Dengan demikian konsep informasi menjadi erat kaitannya dengan pengertian tentang kendala, komunikasi, kontrol, data yang, bentuk, pendidikan, pengetahuan, makna, pemahaman, rangsangan mental yang, pola, persepsi, representasi, dan entropi.

Sistem informasi sering diartikan hanya untuk perangkat lunak yang memproses database dimana ada prosedur prosedur yang diterapkan.

Alter [2] berpendapat untuk sistem informasi sebagai tipe khusus dari sistem kerja. Sistem kerja

adalah suatu sistem di mana manusia dan/atau mesin melakukan pekerjaan dengan menggunakan sumber daya untuk memproduksi produk tertentu dan/atau jasa bagi pelanggan. Sistem informasi adalah suatu sistem kerja yang kegiatannya ditujukan untuk pengolahan (menangkap, transmisi, menyimpan, mengambil, memanipulasi dan menampilkan) informasi [2].

Sistem Informasi merupakan studi dengan referensi kusus yang terdiri dari orang orang, organisasi dan prosedur untuk menggabungkan masukan, proses, filetering dan distribusi data yang bermuara pada informasi bagi pengguna sesuai dengan kebutuhan masing masing. Suatu sistem informasi memiliki Boundary definitif, User, Input, process, Output dan the aforementioned communication networks. [11]

Setiap sistem informasi bertujuan untuk mendukung operasi, manajemen dan pengambilan keputusan. [6] [18] Sistem informasi adalah teknologi informasi dan komunikasi (ICT)

Di mana suatu organisasi menggunakannya dan orang orang berinteraksi menggunakan teknologi ini berdasarkan proses bisnis yang diterapkan. [13]

2.2. Object Oriented Programming(OOP)

Object Oriented Programming (OOP) merupakan model pemrograman yang berbasis pada konsep obyek, diantaranya berisi data, sering dikenal sebagai atribut dan kode, dalam bentuk prosedur, sering dikenal sebagai metode. Sebuah fitur dari obyek adalah bahwa prosedur obyek dapat mengakses dan sering memodifikasi data dari obyek yang saling berhubungan. Dalam OOP, program dirancang dengan membuat obyek yang dapat berinteraksi satu sama lain. [12]

Pemrograman berorientasi obyek didefinisikan menggunakan objek, tetapi tidak semua teknik terkait dan struktur dalam bahasa pemrograman mendukung OOP. [14]

Terkadang obyek-obyek sesuai dengan hal-hal yang ditemukan di dunia nyata. Misalnya, program grafis mungkin memiliki objek seperti "lingkaran", "kotak", "menu". Sebuah sistem belanja online mungkin memiliki objek seperti "keranjang belanja", "pelanggan", dan "produk". [5] Kadang-kadang objek mewakili entitas yang lebih abstrak, seperti objek yang mewakili file terbuka, atau benda yang menyediakan layanan menerjemahkan pengukuran.

Bahasa yang mendukung class hampir selalu mendukung Inheritance. Hal ini memungkinkan Class yang akan diatur dalam hirarki yang mewakili hubungan. Misalnya, Class karyawan mungkin mewarisi dari Class Person. Semua data dan metode yang tersedia untuk Class induk juga muncul di Class anak dengan nama yang sama. Misalnya, Class Orang mungkin mendefinisikan variabel `first_name` dan `last_name` dengan metode `make_full_name()`. Ini juga akan tersedia dalam Class karyawan, yang bisa menambahkan variabel posisi dan gaji. Teknik ini memungkinkan mudah digunakan kembali prosedur

dan data yang sama definisi, selain berpotensi mirroring hubungan dunia nyata dengan cara yang intuitif. Class-class dan subclass sesuai dengan set dan subset di logika matematika. Daripada menggunakan tabel database dan pemrograman subrutin, pengembang memanfaatkan objek, pengguna mungkin lebih akrab dengan Objek dari domain aplikasi mereka [10]

2.3.Web

Situs web [15] adalah kumpulan halaman web yang saling berhubungan, termasuk didalamnya konten multimedia, biasanya diidentifikasi dengan umum nama domain, dan ditampilkan pada setidaknya satu web server. Sebuah situs web dapat diakses melalui public Internet Protocol jaringan (IP), seperti Internet, atau pada jaringan Local Area Network (LAN), dengan referensi sebuah Uniform Resource Locator (URL) yang mengidentifikasi situs. Semua situs yang dapat diakses publik merupakan kolektif World Wide Web (WWW), sementara situs pribadi biasanya bagian dari intranet.

Situs web dinamis merupakan situs web yang secara spesifik didisain agar isi yang terdapat dalam situs tersebut dapat diperbarui secara berkala dengan mudah. Sesuai dengan namanya, isi yang terkandung dalam situs web ini umumnya akan berubah setelah melewati satu periode tertentu. Situs berita adalah salah satu contoh jenis situs yang umumnya mengimplementasikan situs web dinamis.

Web dinamis berbeda dengan web statis, web dinamis membutuhkan hosting dan server yang lebih besar, karena ada proses pengolahan data pada web dinamis. Sementara web statis hanya berisikan kode untuk display saja.

Web server adalah sebuah perangkat lunak yang berfungsi menerima permintaan halaman web dari HTTPS atau HTTP. Agar web server dapat menampilkan halaman web saat ada permintaan dari user, umumnya web server dilengkapi dengan mesin penerjemah bahasa script, serta software manajemen basis data seperti MySQL.

Web dinamis dapat memberikan pengalaman pada pengguna antara lain web yang hidup, dinamis dan interaktif. Konten (teks, gambar, kolom formulir, dll) pada halaman web dapat berubah dalam situasi konteks atau kondisi yang berbeda.

Ada dua cara untuk membuatnya:

1. Menggunakan script di sisi klien untuk mengubah spesifikasi antarmuka pada halaman web, menanggapi input dari mouse atau keyboard pada waktu tertentu. Dalam hal ini perilaku dinamis terjadi pada display.
2. Menggunakan server-side scripting untuk mengubah source page yang ada diantara halaman web, menyesuaikan urutan halaman web atau konten web yang dikirim ke browser. Respon server dapat ditentukan oleh kondisi seperti data dalam yang diposting dalam bentuk HTML, parameter dalam URL,

jenis browser yang digunakan, perjalanan waktu, atau database atau server negara. [15]

2.4.Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya.[7] UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.[7] UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.[7] Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.[7] UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.[7] UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi.[17] UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson.[17] Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi.[17] Penggunaan UML dalam industri terus meningkat.[17] Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.[17] Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia.

Use case diagram digunakan untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem.[1] Use case diagram terdiri atas diagram untuk use case dan actor.[1] Actor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.[1]

Use case merepresentasikan operasi-operasi yang dilakukan oleh actor.[1] Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke use case.[1]

Diagram Class dan diagram Object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti Diagram Interaction. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Collaboration diagram dipakai untuk memodelkan interaksi antar object di dalam sistem.[1] Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan,

collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh object.[1] Class diagram menggambarkan struktur statis class di dalam sistem. class merepresentasikan sesuatu yang ditangani oleh sistem. class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung/menggunakan class yang lain), specialised (satu class merupakan spesialisasi dari class lainnya), atau package (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa class diagram.

III. METODE PENELITIAN

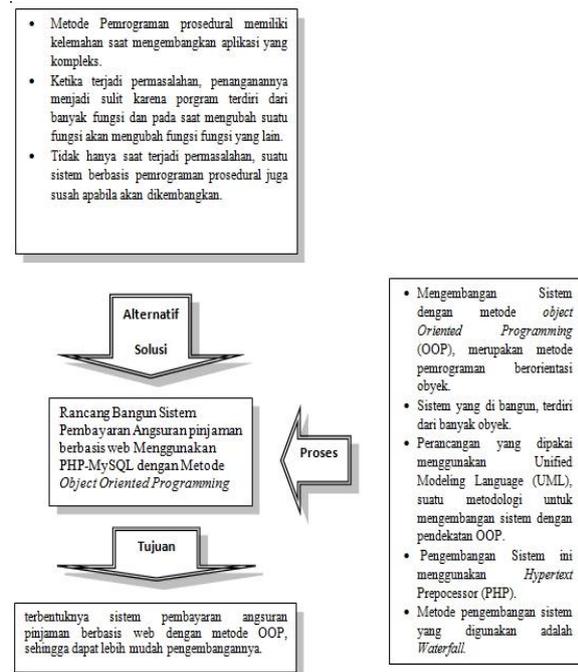
3.1. Lokasi Penelitian

Penelitian di dilakukan di Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.

3.2. Waktu Penelitian

Penelitian ini dilaksanakan pada semester Genap 2016-2017

3.3. Kerangka Berfikir Penelitian

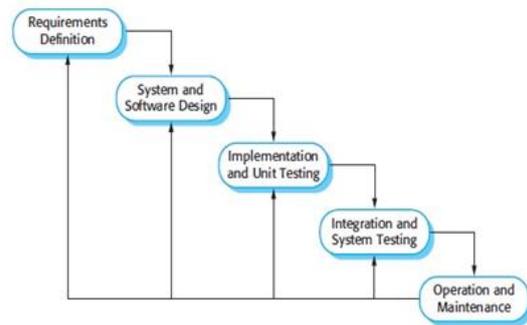


Gambar 1. Kerangka berfikir penelitian

3.4. Metode Pengembangan Sistem

Metode penelitian yang digunakan adalah Rekayasa Sistem Berbasis Komputer berdasarkan prinsip-prinsip rekayasa perangkat lunak (*Software Engineering*) melalui tahapan pengembangan berdasarkan *Waterfall*. Menurut Ian Sommerville, terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition, system and software design, implementation*

and unit testing, integration and system testing, operation and maintenance.



Gambar 2. Metode Pengembangan Sistem *Waterfall* menurut Ian Sommerville

3.4.1. Requirement Analysis and Definition

Pada tahap ini peneliti melakukan analisis data kebutuhan yang telah dikumpulkan, untuk selanjutnya didefinisikan agar dapat dipenuhi oleh sistem yang akan dikembangkan. Fase ini akan menentukan hasil desain. Hal ini sangat penting, mengingat Sistem harus dapat berinteraksi dengan bagian-bagian yang lain seperti *Interface, hardware, database, dan user.*

3.4.2. System and Software Design

Desain perangkat lunak adalah proses yang berfokus pada empat atribut program yaitu: struktur data, asitektur perangkat lunak, representasi interface dan detail (algoritma) prosedural. Desain dikerjakan setelah kebutuhan dikumpulkan dan dianalisis secara lengkap. Peneliti menggunakan *Unified Modelling Language (UML)* untuk membuat desain, di mulai dari *Use Case, Sequence Diagram, Activity Diagram* dan *Class Diagram.*

3.4.3. Implementation and Unit Testing

Desain yang sudah terbentuk diterjemahkan dengan pembuatan kode dalam bahasa pemrograman. Untuk sistem ini, penulis menggunakan *PHP* dengan model *OOP* dan Database *MySQL*, jika desain dilakukan dengan cara yang lengkap, sistem yang di bangun akan sesuai dengan kebutuhan. Sistem yang sudah selesai dibangun langsung diuji melalui pengujian *White Box* dan pengujian *Black Box*

3.4.4. Integration and System Testing

Setelah Sistem selesai dibangun, maka proses pengujian dilakukan pada logika internal untuk memastikan unit unit program sudah diuji. Serelah itu dilakukan pengujian sistem secara keseluruhan untuk memastikan output yang dihasilkan sesuai dengan proses berdasarkan input dari user.

3.4.5. Operation and Maintenance

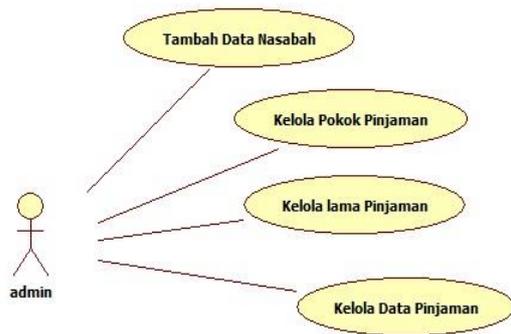
Program yang sudah diterapkan pada lingkungan pengguna akan mengalami penyesuaian atau perubahan

karena adaptasi dengan situasi sebenarnya. Kelemahan dari model waterfall adalah melakukan perubahan setelah diterapkan. oleh karena itu pada proses desain, perlu dilakukan analisis data sesuai kebutuhan pengguna.

IV. HASIL & PEMBAHASAN

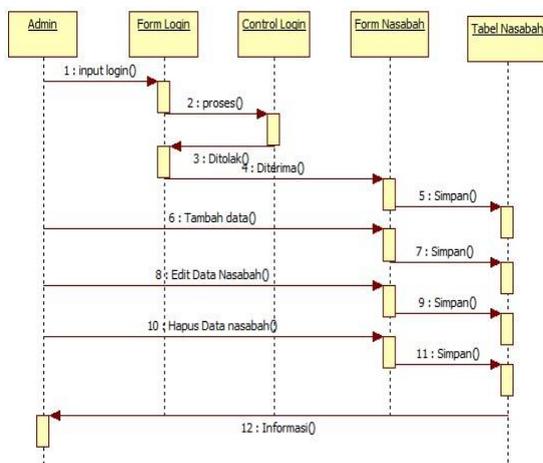
4.1. System and Software Design

Tahap pertama dari software Design ini adalah pembuatan Use Case yang berguna untuk melihat fungsi fungsi apa yang dapat dijalankan oleh suatu aktor



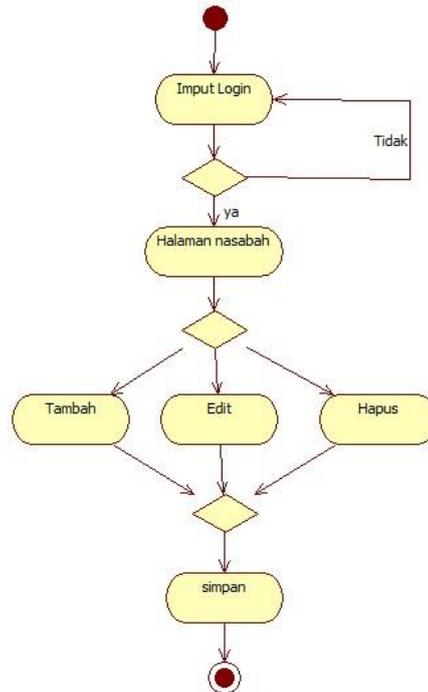
Gambar 3. Use Case Diagram Sistem

Pembuatan Sequence Diagram, digunakan untuk menunjukkan interaksi yang terjadi antara objek dengan sistem. Diagram ini merupakan pandangan dinamis terhadap sistem yang terjadi.



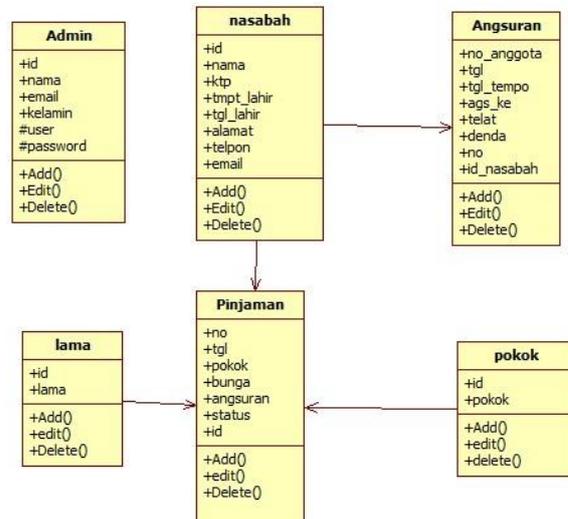
Gambar 4. Sequence Diagram Tambah Data Nasabah

Pembuatan Activity Diagram. Merupakan diagram yang menggambarkan aliran kerja atau proses bisnis suatu sistem.



Gambar 5. Activity Diagram Tambah Data Nasabah

Pembuatan Class Diagram, untuk menggambarkan class, atribut, fungsi dan bentuk relasinya.



Gambar 6. Class Diagram

4.2. Hasil

- a. *Class, Object, Method* dalam PHP OOP Merupakan cetakan atau properti yang mendefinisikan *variable* dan *method* dan berkalu umum untuk semua obyek. Atribut dari *class* bisa diturunkan ke obyek namun *Object* bisa memiliki *method* yang berbeda.

```

File Edit Format View Help
<?php
class Binatang {
    public $jenis;
    public $earna;

    function bersuara() {
        echo "guk guk guk. ini adalah binatang ac
    >jenis";
    }

    function statuswarna() {
        if ($this-> warna = "Hitam" )
            echo $status = 'Jinak' ;
        else
            echo $status = 'Liar';
        return $status;
    }
}

$objbinatang = new binatang();

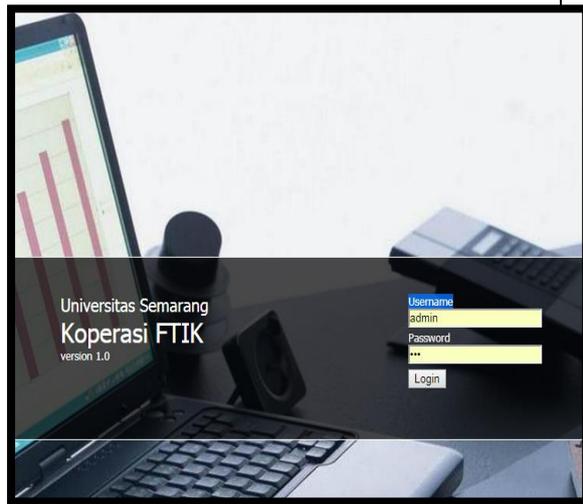
$objbinatang->jenis = 'Anjing';
$objbinatang->warna = 'Coklat';

echo "jenis : $objbinatang->jenis <br />";
echo "Warna : $objbinatang->warna";
?>
    
```

Gambar 7. Class dalam PHP OOP

b. Halaman Home

Halaman home berisi menu menu yang dimiliki sistem, user dalam hal ini admin dapat mengakses menu sesuai dengan yang dibutuhkan.



Gambar 8. Halaman Home

c. Menu Sistem

Menu sistem ini berisi fungsi fungsi dari sistem, mulai dari data nasabah, data pokok pinjaman, data lama pinjaman, data pinjaman dan laporan pinjaman.



Gambar 9. Menu Sistem

4.2.3. Pengujian

Dalam pengujian ini, peneliti akan menggunakan pengujian *Blackbox*. Merupakan metode pengujian dengan mengamati hasil dari data uji dan fungsional sebuah sistem.

No	Fungsi	Butir Uji	Hasil Uji
1	Input, edit, hapus data nasabah	Isi form dengan data nasabah baru. Hasilnya data nasabah akan bertambah dengan data yang baru saja diinputkan. Demikian juga bila dilakukan perubahan (<i>edit</i>) dengan menekan tombol fungsi <i>edit</i> dan menekan tombol <i>save</i> setelah melakukan perubahan, maka data akan berubah. Biladilakukan penghapusan (<i>Delete</i>) data dengan menekan tombol hapus, maka data akan terhapus	berhasil
2	Transaksi data angsuran	Apabila terjadi proses transaksi pembayaran angsuran, setelah proses selesai dan lakukan penyimpanan dengan menekan tombol <i>save</i> , maka data akan tercatat dan akan mengurangi jumlah angsuran nasabah.	berhasil
3	Cetak laporan	Menu laporan pinjaman berfungsi akan mencetak laporan data pembayaran semua nasabah. Bila menu ini diklik maka akan muncul laporan data nasabah yang siap untuk dicetak baik dalam bentuk pdf atau <i>paper</i>	berhasil

V. SIMPULAN DAN SARAN

a. Kesimpulan

Kesimpulan dari penelitian ini adalah

1. Terbentuknya sistem pembayaran angsuran online menggunakan *PHP OOP* dan metode pengembangan sistem *waterfall* dengan fungsi fungsi sistem yang berjalan sesuai perencanaan .
2. Pemrograman PHP dengan metode OOP mendukung konsep MVC (*Model, View, Controller*). Sehingga programmer dapat lebih mudah mengembangkan aplikasi menjadi lebih kompleks.

b. Saran

Saran dari penulis adalah sebagai berikut:

1. Aplikasi yang dikembangkan dalam penelitian ini masih sederhana, perlu ditambahkan fungsi fungsi baru agar lebih memaksimalkan penerapan metode OOP dalam pemrograman PHP.
2. Pengembangan aplikasi masih memakai metode PHP OOP, untuk kedepan perlu diterapkan konsep MVC, sehingga penyelesaiannya bisa lebih cepat.

DAFTAR PUSTAKA

- [1] AZIZ, FARID , M. Kom., Ir. *Object Oriented Programming Php* 5, halaman 118. Elex Media Komputindo.
- [2] Alter, S. *The Work System Method: Connecting People, Processes, and IT for Business Results*. Works System Press, CA.
- [3] A short overview is found in: Luciano Floridi (2010). *Information - A Very Short Introduction*. Oxford University Press. ISBN 0-19-160954-4. *The goal of this volume is to provide an outline of what information is...*
- [4] Biago Bosson and Massimo Cirasino, *"The Oversight of the Payment Systems: A Framework for the Development and Governance of Payment Systems in Emerging Economies"*The World Bank, July 2001, p.7.
- [5] Booch, Grady (1986). *Software Engineering with Ada*. Addison Wesley. p. 220. ISBN 978-0805306088. *Perhaps the greatest strength of an object-oriented approach to development is that it offers a mechanism that captures a model of the real world.*
- [6] Bulgacs, Simon (2013). *"The first phase of creating a standardised international technological implementation framework/software application"*. *Int. J. Business and Systems Research* 7 (3): 250. - doi:10.1504/IJBSR.2013.055312. Retrieved 11/2/15. *Check date values in: |access-date= (help)*
- [7] David M. Kroenke, *Database Processing* Jilid 1 edisi 9, halaman 60. Erlangga.
- [8] D'Atri A., De Marco M., Casalino N. (2008). *"Interdisciplinary Aspects of Information Systems Studies"*, Physica-Verlag, Springer, Germany, pp. 1-416, doi:10.1007/978-3-7908-2010-2 ISBN 978-3-7908-2009-6
- [9] <http://www.merriam-webster.com/dictionary/information>
- [10] Jacobsen, Ivar; Magnus Christerson; Patrik Jonsson; Gunnar Overgaard (1992). *Object Oriented Software Engineering*. Addison-Wesley ACM Press. pp. 43-69. ISBN 0-201-54435-0.
- [11] Jessup, Leonard M.; Joseph S. Valacich (2008). *Information Systems Today* (3rd ed.). Pearson Publishing. Pages & Glossary p. 416
- [12] Kindler, E.; Krivy, I. (2011). *"Object-Oriented Simulation of systems with sophisticated control"*. *International Journal of General Systems*: 313-343.
- [13] Kroenke, D M. (2008). *Experiencing MIS*. Prentice-Hall, Upper Saddle River, NJ
- [14] Michael Lee Scott, *Programming language pragmatics*, Edition 2, Morgan Kaufmann, 2006, ISBN 0-12-633951-1, p. 470. *Lists encapsulation, inheritance, and dynamic dispatch.*
- [15] Nelson, Anne Fulcher, and Nelson, William Harris Morehead. (2001). *Building Electronic Commerce with Web Database Constructions*. Boston, MA: Addison Wesley.
- [16] *"Payment Systems: Design, Governance and Oversight"*, edited by Bruce J. Summers, Central Banking Publications Ltd, London, 2012, p.3.
- [17] Rama, Jones, *Sistem Informasi Akuntansi* halaman 78. Salemba.
- [18] SEI Report, *"Glossary" at the Wayback Machine* (archived September 3, 2007). *"What is a Payment System?"*. Federal Reserve Bank of New York. 13 Oc 2000. Retrieved 23 July 2015. *Check date values in: |date= (help)*.