

A. Latar Belakang

Ibu Mushodah memiliki usaha kredit perorangan yang memberi kredit berupa alat – alat rumah tangga dengan pembayaran cicilan atau sering disebut dengan istilah mindring. Ibu Mushodah bertempat tinggal di desa Gaji, Kecamatan Guntur, Kabupaten Demak. Setiap hari minggu dan senin Ibu Mushodah berkeliling ke kediaman para nasabah untuk menagih cicilan yang harus dibayarkan. Lebih dari 15 tahun usaha ini berjalan dan jumlah nasabahnya lebih dari 600 orang yang tersebar di Desa Gaji dan Banjarejo.

Dalam menjalankan usahanya Ibu Mushodah hanya mencatat semua data yang berkaitan dengan nasabah kedalam sebuah buku. Beberapa buku catatan harus dibawa setiap kali menagih. Proses pengolahan data masih dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama untuk mencari data dan kemungkinan data ganda atau kehilangan data sangat besar. Hal tersebut menjadi acuan untuk membuat aplikasi kredit perorangan berbasis android yang nantinya dapat diakses langsung oleh Ibu Mushodah untuk menunjang kegiatan usahanya.

Aplikasi kredit perorangan berbasis android menjadi solusi untuk memudahkan proses kelola data nasabah dan pembayaran cicilan menggunakan sistem terkomputerisasi secara *online* yang dapat diakses kapanpun. Semua kegiatan mengolah data secara manual akan lebih baik jika digantikan dengan sistem yang terkomputerisasi. Dengan adanya masalah tersebut maka Ibu Mushodah membutuhkan teknologi atau aplikasi yang menjamin sumber data, kuantitas dan waktu yang lebih efisien. Sesuai uraian diatas, maka penulis tertarik untuk membahas lebih dalam mengenai pengolahan data kredit perorangan milik Ibu Mushodah dengan judul **“APLIKASI KREDIT PERORANGAN BERBASIS ANDROID”**.

B. Rumusan Masalah

Dari uraian latar belakang yang telah dibuat dapat di ambil rumusan masalah yaitu:

Bagaimana membangun sebuah aplikasi kredit perorangan berbasis android dengan data yang terintegrasi dengan *server*.

C. Batasan Masalah

Penulis memiliki batasan masalah untuk menghindari meluasnya pembahasan materi agar lebih terarah. Adapun batasan masalah yang akan dibahas yaitu:

1. Aplikasi *frontend* dibangun menggunakan Ionic Framework.
2. *Backend* menggunakan Laravel PHP Framework.
3. RDBMS menggunakan MySQL.
4. Aplikasi hanya digunakan oleh perorangan.
5. Model pengembangan sistem menggunakan *waterfall*.
6. Detail sistem.
 - Tidak menggunakan sistem bunga *bank*.
 - Tidak menggunakan jaminan.
 - Penarikan cicilan hanya seminggu sekali.
 - Jika nasabah tidak membayar cicilan maka jatuh tempo mundur 7 hari.

D. Tujuan Tugas Akhir

Tujuan yang akan dicapai dari tugas akhir ini adalah membuat aplikasi kredit perorangan berbasis android yang mampu memudahkan pekerjaan Ibu Mushodah dalam mengelola data nasabah dan pembayaran cicilan agar lebih efisien.

E. Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah:

1. Bagi Penulis

Dapat menerapkan teori maupun praktek yang telah didapatkan selama dibangku perkuliahan dan dapat mengukur kemampuan penulis dalam membuat program aplikasi bagi masyarakat.

2. Bagi Ibu Mushodah

Dapat mempermudah pekerjaan karena tidak perlu membawa buku – buku catatan dan mengurangi resiko kehilangan data.

3. Bagi Pembaca

Bagi pembaca dapat digunakan sebagai sumber informasi atau referensi bagi penelitian lebih lanjut.

F. Tinjauan Pustaka

1. Kredit

Istilah kredit berasal dari bahasa Yunani, *Credere* yang berarti kepercayaan (*Truth* atau *Faith*). Oleh karena itu dasar dari kredit adalah adanya kepercayaan. Seseorang atau badan usaha yang memberikan kredit (kreditur) memberikan kepercayaan bahwa penerima kredit (debitur) di masa mendatang akan sanggup memenuhi segala sesuatu yang telah disepakati kedua pihak. Kegiatan perkreditan melibatkan beberapa pihak, diantaranya adalah bank, debitur, otorita atau pemerintah, dan masyarakat. Tujuan kredit pun berbeda – beda tergantung pada pihak-pihak tersebut (Saraswati, 2012).

2. PHP

PHP *Hypertext Processor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP dapat diintegrasikan dengan

HTML, JavaScript, JQuery, Ajax. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (*Situs Personal*). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web* (Hidayatullah dan Jauhari, 2014).

3. RDBMS MySQL

MySQL adalah *Database Management System* (DBMS) yang sangat banyak digunakan oleh para pemrogram aplikasi *web*. Kelebihan MySQL adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna. MySQL juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya lebih mudah. DBMS adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang integrasinya seperti membuat, menghapus, menambah dan memodifikasi basis data serta mampu berkomunikasi dengan program aplikasi lain misalnya dimungkinkan untuk mengakses basis data MySQL menggunakan aplikasi yang dibuat menggunakan PHP (Hidayatullah dan Jauhari, 2014).

4. Laravel Framework

Laravel adalah kerangka aplikasi *web* dengan sintaks ekspresif dan elegan. Pengembangan aplikasi harus menjadi pengalaman yang kreatif dan menyenangkan agar benar – benar memuaskan. Laravel berupaya menghilangkan kesulitan pengembangan dengan mengurangi tugas umum yang digunakan di sebagian besar proyek – proyek *web*, seperti otentikasi, *routing*, sesi, dan *caching* (Brujah, 2104).

Laravel bertujuan untuk membuat proses pengembangan aplikasi menyenangkan untuk para *developer* tanpa mengorbankan fungsionalitas dari suatu aplikasi. Pengembangan terbaik akan menghasilkan kode yang baik. Untuk mencapai tujuan ini, kami telah berusaha untuk menggabungkan yang terbaik dari apa yang telah kita lihat dalam kerangka web lain, termasuk kerangka kerja yang diimplementasikan dalam bahasa lain seperti Ruby on Rails, ASP.NET MVC dan Sintara (Brujah, 2014).

Laravel sangat mudah diakses, powerful, menyediakan alat – alat bantu yang hebat yang diperlukan untuk membangun aplikasi yang besar dan kuat (Brujah, 2014).

5. Ionic Framework

Ionic adalah sebuah *frontend* SDK yang bagus, *open source* dan untuk mengembangkan aplikasi – aplikasi *mobile* dengan HTML5. Ionic menyediakan komponen HTML, CSS dan JS yang teroptimalisasi untuk *mobile* seperti *gesture* dan *tool* untuk membangun aplikasi yang sangat interaktif (Ravulavaru, 2015).

6. Perancangan dan Analisa Sistem

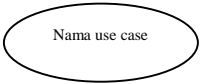


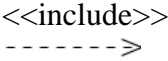
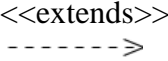
Perancangan sistem pada tugas akhir ini dilakukan dengan pembuatan perancangan dengan menggunakan *UML (Unified Modeling Language)*. *UML* adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat


lunak (Rosa, 2013). *UML* mendefinisikan diagram – diagram grafis berikut ini:

a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa dan Shalahuddin, 2013). Simbol yang digunakan dalam *use case diagram* adalah :

Tabel 1. Simbol-simbol dalam *Use Case Diagram* (Rosa, 2013)

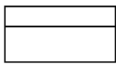



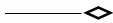
SIMBOL	NAMA	DESKRIPSI
	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesa antar unit atau aktor.
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	<i>Generalization</i>	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<i>Include</i>	<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

	<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
---	--------------------	---

b. *Class Diagram*

Class Diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa dan Shalahuddin, 2013). Simbol yang digunakan dalam *Class Diagram* adalah sebagai berikut :

Tabel 2. Simbol-simbol dalam *Class Diagram* (Rosa, 2013)

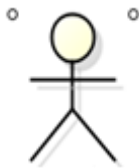

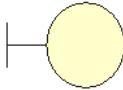
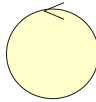
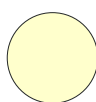

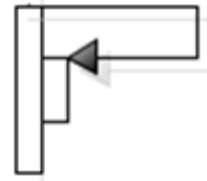
SIMBOL	NAMA	DESKRIPSI
	<i>Class</i>	Kelas pada struktur sistem.
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
	<i>Association</i> / asosiasi	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Directed Association</i> / asosiasi berarah	Relasi antar kelas dengan makna kelas yang satudigunakan oleh kelas yang lain.
	<i>Aggregation</i> / agregasi	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

c. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message*

yang dikirimkan dan diterima antar objek. Menggambar diagram sekuen harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa, 2013). *Sequence diagram* dapat dinotasikan seperti gambar dibawah ini :


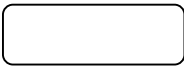
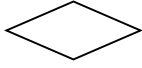


Tabel 3. Simbol – simbol dalam *Sequence Diagram* (Nugroho, 2010).

SIMBOL	NAMA	DESKRIPSI
	<i>Actor</i>	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan istem yang dikembangkan.
	<i>Lifeline</i> / garis waktu	Menyatakan kehidupan suatu objek.
	<i>Boundary Class</i>	Menyatakan form pengisian.
	<i>Control Class</i>	Menyatakan penghubung atau proses.
	<i>Entity Class</i>	Menyatakan tabel atau <i>database</i> .
	<i>Synchronous message</i>	Mengaktifkan sebuah proses sampai selesai, kemudian baru bisa menyampaikan <i>message</i> baru.
	<i>Message to self</i>	Suatu hasil kembalian sebuah operasi dan berjalan pada objek itu sendiri.

d. *Activity Diagram*

Activity Diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa, 2013). Berikut simbol – simbol yang digunakan dalam *Activity Diagram*:

Tabel 4. Simbol-simbol dalam *Activity Diagram* (Rosa, 2013).

SIMBOL	NAMA	DESKRIPSI
	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Decision / Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Join / Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

G. Metodologi Penelitian

Metodologi penelitian sangat penting dalam suatu penelitian karena kesimpulan yang diambil dapat dipengaruhi oleh metodologi penelitian yang dipakai. Metodologi penelitian juga merupakan suatu cara atau prosedur yang digunakan untuk mengumpulkan, mengolah dan menganalisa data. Metode yang dilakukan penulis diantaranya:

1. Sumber Data

a. Data Primer

Data primer merupakan data yang diperoleh secara langsung dari objek penelitian yaitu dari bagian yang berhubungan secara langsung dengan praktek kerja Ibu Mushodah.

b. Data Sekunder

Data sekunder merupakan data yang diperoleh secara tidak langsung, yaitu data diperoleh dari berbagai literatur dan buku – buku di perpustakaan maupun artikel dari internet yang berhubungan dengan pembuatan aplikasi serta pelengkap data yang akan membantu penyelesaian laporan tugas akhir.

2. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan penulis antara lain sebagai berikut:

a. Studi Lapangan

Dengan metode ini data diperoleh dengan mengadakan pengamatan secara langsung pada objek yang diteliti, studi lapangan ini dilakukan dengan cara :

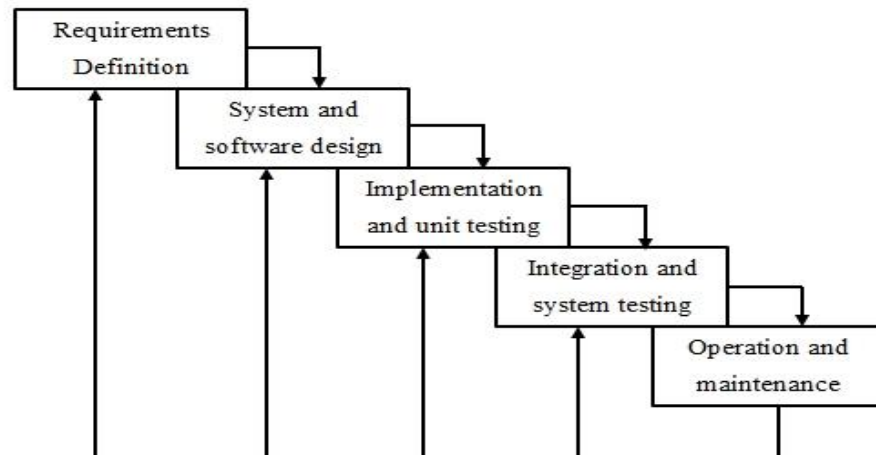
1. Wawancara; melakukan tanya jawab kepada Ibu Mushodah dan beberapa nasabah yang sudah terbiasa melakukan kredit.
2. Observasi; ikut serta membeli dan mengirim barang pesanan nasabah, serta mengamati proses akad perjanjian.

b. Studi Kepustakaan

Melakukan pengumpulan data dengan membaca dan mengambil dari sejumlah buku di perpustakaan, jurnal dan sumber lainnya yang berhubungan dengan masalah yang diamati.

3. Metode Pengembangan Sistem

Dari data yang terkumpul kemudian dilakukan pengembangan sistem menggunakan model Waterfall seperti terlihat pada Gambar 1.



Gambar 1. *Waterfall* (Sommerville, 2011).

a. *Requirement Definition*

Mengumpulkan data tentang nasabah, seperti nama, alamat dan lain – lain. Serta mencatat alur kegiatan dalam bertransaksi.

b. *System and Software Design*

Membuat rancangan *database* untuk menampung semua data dari nasabah, merancang antar muka untuk pengguna aplikasi.

c. *Implementation and Unit Testing*

Menerapkan semua data yang terkumpul mulai dari kebutuhan dan desain kedalam bahasa pemrograman, membuat fungsi – fungsi, menguji setiap fungsi agar berjalan sesuai skenario.

d. *Integration and System Testing*

Penggabungan seluruh komponen *frontend* dan *backend*. kemudian menguji semua sistem agar berkomunikasi dengan baik.

e. *Operation and Maintenance*

Mengoperasikan aplikasi kedalam pekerjaan Ibu Mushoah, dan melakukan pemeliharaan sistem, seperti penyesuaian atau perubahan sistem karena adaptasi dengan situasi sebenarnya.

H. Jadwal Pelaksanaan

Berikut ini akan ditampilkan jadwal pelaksanaan tugas akhir.

ID	Task Name	Start	Finish	Duration	Okt 2015			Nov 2015				Des 2015				Jan 2016				Feb 2016			
					11/10	18/10	25/10	1/11	8/11	15/11	22/11	29/11	6/12	13/12	20/12	27/12	3/1	10/1	17/1	24/1	31/1	7/2	14/2
1	Studi Literatur	08/10/2015	02/11/2015	18d																			
2	Proposal TA	03/11/2015	13/11/2015	9d																			
3	Pengumpulan Data	03/11/2015	30/11/2015	20d																			
4	Analisa	13/11/2015	01/12/2015	13d																			
5	Perancangan Aplikasi	24/11/2015	29/12/2015	26d																			
6	Implementasi dan Pengujian	22/12/2015	22/01/2016	24d																			
7	Penyusunan Laporan	13/11/2015	19/02/2016	71d																			
8	Ujian / Sidang TA	22/02/2016	29/02/2016	6d																			

Gambar 2. jadwal pelaksanaan tugas akhir.