

Rancang Bangun Sistem Informasi emberkasan Arsip Debitur Menggunakan Metode Hashing (Studi Kasus : PT. Bank Mandiri (Persero) Tbk Jember) (Design of Information Systems Archive Filing Debtor Uses Hashing Method (Case Study: PT Bank Mandiri (Persero) Tbk Jember))

Ayu Priyanti, Dwiretno Istiyadi S, Windi Eka Yulia R
Program Studi Sistem Informasi Universitas Jember (UNEJ)
Jln. Kalimantan 37, Jember 68121
E-mail: istiyadi.s@gmail.com

Abstrak

PT. Bank Mandiri (Persero) Tbk adalah bank yang berkantor pusat di Jakarta, dan merupakan bank terbesar di Indonesia dalam hal aset, pinjaman, dan deposit. Banyaknya calon debitur yang melakukan proses peminjaman dana di PT. Bank Mandiri (Persero) Tbk Jember membuat pegawai pada bagian peminjaman yaitu *Customer Loan Business Office (CLBO)* kesulitan dalam proses penyimpanan berkas pinjaman debitur, yang mana saat ini mereka melakukannya secara manual. Sistem Informasi Pemberkasan Arsip Debitur Menggunakan Metode *Hashing* dapat membantu menentukan nomor loker untuk penyimpanan arsip pinjaman debitur ke dalam loker yang telah disediakan. Sistem Pemberkasan Arsip Debitur diimplementasikan dalam bentuk penggunaan *web*, sehingga dapat diakses secara mudah oleh *user*. Aplikasi yang dikembangkan menggunakan bahasa pemrograman PHP dan didukung oleh penggunaan *framework Code Igniter*.

KataKunci: Arsip, Debitur, *Filing System*, Metode *Hashing*

Abstract

PT. Bank Mandiri (Persero) Tbk is a bank headquartered in Jakarta, and is the largest bank in Indonesia in terms of assets, loans, and deposits. The number of borrowers who do the process of borrowing funds in PT. Bank Mandiri (Persero) Tbk Jember make employees at section of borrowing, Customer Loan Business Office (CLBO), difficulties in the debtor's loan file storage, which they currently do it manually. Archive Filing Debtor Information System Using Hashing method can help determine the number archival storage locker for loan borrowers into lockers provided. Filing System Archive Debtor implemented in the form of web use, so it can be accessed easily by the user. Applications developed using the PHP programming language and supported by the use of Code Igniter framework.

Keyword : *Archive, The Debtor, Filing System, Hashing Method*

Pendahuluan

Debitur adalah pihak yang berhutang kepada pihak lain, biasanya dengan menerima sesuatu dari lembaga/perorangan yang dijanjikan debitur untuk dibayar kembali pada masa yang akan datang [2]. Dalam melakukan proses peminjaman debitur diwajibkan untuk mengumpulkan beberapa persyaratan peminjaman dana yang merupakan dokumen penting yang berisikan arsip-arsip jaminan dari peminjaman yang mereka lakukan. Arsip adalah kumpulan warkat atau dokumen yang disimpan secara sistematis agar setiap kali diperlukan dapat dengan cepat ditemukan kembali [6]. Arsip atau dokumen yang akan disimpan merupakan dokumen penting sehingga dibutuhkan proses penyimpanan dokumen tersebut yang nantinya apabila dokumen atau arsip tersebut diperlukan

kembali dapat didapatkan dengan mudah. *Filing system* adalah sistem penyimpanan dan penemuan kembali informasi yang terdiri dari aspek sistem seperti: lokasi fisik, metode klasifikasi dan pengidekan, pengaturan dan penataan berkas, prosedur pemberkasan, peralatan dan perlengkapan, pelacakan berkas, teknologi yang digunakan dalam implementasi sistem [1]. Metode yang telah digunakan dalam melakukan pengarsipan dan pencarian data salah satunya yaitu metode *hashing*.

Dalam penelitian sebelumnya dengan judul "Aplikasi Tabel *Hash* Dalam Pengarsipan dan Pencarian Data" mengimplementasikan metode *hashing* dengan menggunakan tabel *hash* dalam penyimpanan data yang dilakukan yaitu dengan mengganti nilai huruf menjadi nilai ASCII terlebih dahulu sebelum melakukan modulus dari nilai *integer* pada data. Sehingga proses pengarsipan dan pencarian data dapat mempersingkat waktu yang dibutuhkan.

Pada perusahaan, angka peminjaman terus meningkat sehingga mengakibatkan semakin banyak berkas debitur yang harus pegawai pada bagian CLBO arsipkan. Sedangkan untuk saat ini pengarsipan berkas pinjaman debitur dan penentuan nomor loker berkas pinjaman debitur pada bagian CLBO dilakukan secara manual, dimana pencarian berkas yang diperlukan oleh pegawai memerlukan waktu yang cukup lama dan sangat tidak efektif dalam melakukan penyimpanan dan pencarian berkas dari setiap debitur.

Berdasarkan permasalahan tersebut diatas, maka diperlukan suatu Sistem Pemberkasan Arsip Debitur yang dapat melakukan penyimpanan data dan pengkodean nomor loker untuk setiap arsip debitur di bagian CLBO. Adapun fitur unggulan dari Sistem Informasi Pemberkasan Arsip Debitur yaitu dengan memberikan informasi tentang nomor loker yang nantinya digunakan sebagai tempat penyimpanan arsip debitur sehingga dapat memudahkan pegawai dalam melakukan pencarian arsip debitur secara cepat dengan waktu yang lebih efektif.

Sistem Informasi Pemberkasan Arsip debitur ini dibuat dengan menggunakan model pengembangan perangkat lunak *Prototype*. Metode yang digunakan dalam pengarsipan arsip debitur yaitu dengan menggunakan metode *hashing*. Metode *hashing/scatter storage* meliputi suatu perhitungan aritmatika pada nilai kunci untuk menghasilkan satu bilangan bulat/*integer* yang merupakan alamat relatif dimana kunci disimpan [3].

Metode Penelitian

Pada tahap pengumpulan data teknik yang digunakan yaitu studi lapangan (*Field Research*) dan studi kepustakaan (*Library Research*). Penelitian lapang (*Field Research*) ini dilakukan dengan cara bertemu langsung dengan objek yang diteliti yaitu pegawai CLBO di PT. Bank Mandiri (Persero) Tbk cabang Jember dengan menggunakan dua metode yaitu observasi dan wawancara. Sedangkan penelitian dengan teknik studi kepustakaan (*Library Research*) dilakukan dengan cara mencari informasi dari buku-buku atau sumber tulisan lainnya yang berhubungan dengan rumusan masalah yang akan kita bahas nantinya dengan tujuan memperoleh data yang bersifat teori.

Untuk konsep sistem yang akan digunakan yaitu dengan menggunakan konsep sistem OOP. Pengimplementasian sistem ini dengan menggunakan bahasa pemrograman PHP, CSS, dan *Javascript*. PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai *scriptserver-side* dalam pengembangan web yang disisipkan pada dokumen HTML [4]. Sedangkan manajemen data pada sistem ini menggunakan DBMS *MySQL*. *MySQL* adalah sebuah *management system database server* yang mampu menangani beberapa *User*, yaitu mampu menangani instruksi sekaligus dari beberapa *User* dalam satu waktu [5]. Dengan kita memakai *MySQL* maka *database* mampu mengirim dan menerima data dengan cepat dan dapat digunakan oleh multi *user*.

Hasil Penelitian

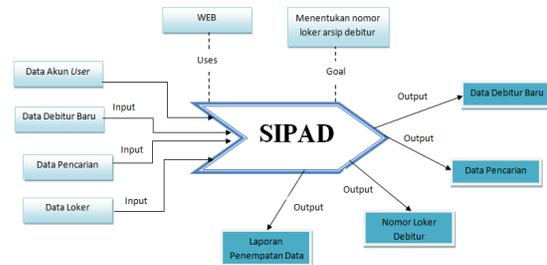
Desain dan Perancangan Sistem

Desain sistem menjelaskan tentang tahapan dimana sistem yang akan dibuat akan diaplikasikan dalam bentuk dokumen sistem yang meliputi *Business Process*, *Usecase Diagram*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram* dan

Entity Relationship Diagrams (ERD).

1. Business Process

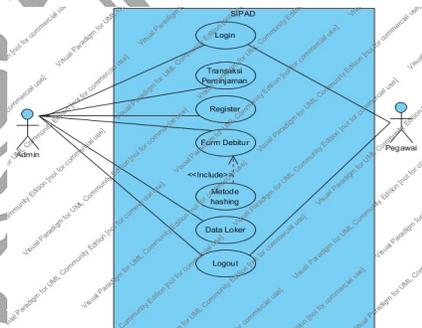
Business Process pada Sistem Informasi Pemberkasan Arsip Debitur menjelaskan tentang proses sistem yang akan dibangun seperti masukan dan keluaran, penggunaan sumber daya, dan tujuan. Penjelasan mengenai *business process* dapat dilihat pada Gambar 1.



Gambar 1. *Business Process* Sistem Informasi Pemberkasan Arsip Debitur (Sumber: Hasil Analisis, 2014)

2. Usecase Diagram

Pembuatan *usecase diagram* menjelaskan tentang apa saja yang dapat dilakukan dalam Sistem Informasi Pemberkasan Arsip Debitur serta menggambarkan kebutuhan fungsional dari sistem tersebut. Berikut adalah detail *Usecase Diagram* dapat dilihat pada Gambar 2.



Gambar 2. *Usecase Diagram* Sistem Informasi Pemberkasan Arsip Debitur (Sumber: Hasil Analisis, 2014)

3. Fitur Utama

Fitur utama yang akan dibahas yaitu fitur Transaksi Peminjaman. Fitur ini merupakan fitur lanjutan dari proses *input* data diri dan data pinjaman debitur yang akan ditampilkan pada tabel transaksi peminjaman yang ada pada menu transaksi peminjaman. Pada fitur ini metode *hashing* digunakan untuk menentukan nomor loker sebagai tempat penyimpanan arsip debitur yang melakukan peminjaman dana pada perusahaan.

a. Sequence Diagram Transaksi Peminjaman

Sequence diagram pada Sistem Informasi Pemberkasan Arsip Debitur menggambarkan tentang urutan proses yang dapat dilakukan oleh sistem untuk mencapai tujuan dari *usecase* yang terjadi antar *class*, operasi apa saja yang terlibat, dan informasi yang diperlukan oleh masing-masing operasi. *Sequence diagram* pada fitur Transaksi Peminjaman dapat dilihat pada Gambar 3.

Pembahasan

Implementasi Metode Hashing

Pada pembahasan akan dijelaskan mengenai pengkodean sistem dalam implementasi dari metode *hashing* pada transaksi peminjaman. Pada Gambar 8 merupakan gambar yang menjelaskan kode program yang digunakan

ketika sistem menjalankan menu transaksi peminjaman dan terdapat juga kode program untuk perhitungan metode *hashing*.

```

public function form_pinjaman()
{
    $config = array(
        array(
            'field' => 'jumlahCicilan',
            'tabel' => 'jumlahCicilan',
            'rules' => 'required',
        ),
        array(
            'field' => 'jumlahCair',
            'tabel' => 'jumlahCair',
            'rules' => 'required',
        ),
        array(
            'field' => 'tempo',
            'tabel' => 'tempo',
            'rules' => 'required',
        ),
        array(
            'field' => 'ket',
            'tabel' => 'ket',
        ),
    );
    $this->form_validation->set_rules($config);
    if ($this->form_validation->run() == true) {
        $cek = $this->input_debitur->cekNoJaminan();
        if ($cek->jumlah%2==1) {
            setcookie('tesh'," NO Jaminan Masih Dalam Masa Peminjaman",time()+10,1);
            redirect('tampilanform_pinjaman');
        }
        $jumlahLoker = $this->input_debitur->getJumlahLokerKosong();
        if ($jumlahLoker==0) {
            setcookie('tesh'," Loker Penuh",time()+10,1);
            redirect('tampilanform_pinjaman');
        }
        $no_arsip = $this->input_debitur->insertTransaksi(peminjaman);
        $noLoker = ($no_arsip+no_arsip % $jumlahLoker) + 1;
        $this->input_debitur->setLokerArsip($noLoker, $no_arsip+no_arsip);
        $noLoker = $this->input_debitur->getNoLoker($no_arsip+no_arsip);
        $noLoker = $noLoker+noLoker;
        if ($this->upload->is_uploaded_file($noLoker) {
            mkdir('upload/' . $noLoker . '/');
        }
        $this->load->library('upload');
        $name = array(
            "1" => "AplikasiPeminjaman",
            "2" => "BuktiJaminan",
            "3" => "BuktiPenyertaan");
        for ($i = 1; $i <= 3; $i++) {
            $config['upload_path'] = 'upload/' . $noLoker . '/';
            $config['allowed_types'] = 'pdf';
            $config['max_size'] = '10240';
            $config['file_name'] = $name[$i];
            $config['overwrite'] = true;
            $this->upload->initialize($config);
            $this->upload->do_upload($name[$i]);
        }
        redirect('tampilaninfo_pinjaman');
    }
    $data_peminjaman($diPinjaman) = $this->input_debitur->ddiPinjaman();
    $data_peminjaman('content') = form_pinjaman();
    $this->load->view('template', $data_peminjaman);
}

```

Gambar 8. Baris Program Transaksi Peminjaman Pada Controller (Sumber: Hasil Analisis, 2014)

Pada menu transaksi peminjaman, sistem akan menjalankan fungsi dari metode *hashing*. Langkah 1: admin melakukan *input* data yang berkaitan dengan data pinjaman yang diajukan oleh debitur pada perusahaan. Beberapa *field* yang tersedia pada *form* pinjaman meliputi jenis pinjaman, jumlah pencairan, tanggal pinjaman, tanggal jatuh tempo, jumlah cicilan, keterangan pinjaman, dan berkas yang digunakan dalam pengajuan pinjaman dana oleh debitur. Langkah 2: Setelah proses validasi pengisian *form* benar, maka sistem akan melakukan pengecekan nomor jaminan yang debitur gunakan sebagai syarat dari proses peminjaman pada perusahaan. Ketika nomor jaminan masih berada dalam masa peminjaman maka nomor jaminan tersebut tidak dapat digunakan oleh debitur lainnya yang ingin melakukan peminjaman dana pada

perusahaan. Sehingga nantinya ketika admin memasukkan nomor jaminan yang masih berada dalam masa peminjaman, sistem akan menampilkan *message warning* bahwa nomor jaminan dalam masa pinjaman. Langkah 3: Setelah proses pengecekan nomor jaminan telah selesai dan nomor jaminan belum terpakai maka sistem akan melanjutkan pada proses pengambilan jumlah loker kosong. Kode program `getJumlahLokerKosong()` yang ada pada *class* model. Kode program yang ada pada *class* model yaitu `getJumlahLokerKosong()` dapat dilihat pada Gambar 9 berikut.

```

//proses pengambilan no loker kosong
function getJumlahLokerKosong()
{
    $this->db->select("count(no_loker) as jumlah");
    $this->db->from('data_loker');
    $this->db->where('jenis_loker', $this->input->post('id_pinjaman'));
    $this->db->where_not_in('no_loker',
        "(select t1.no_loker from (select * from loker where status='m') t1 join
        (select * from loker where status='k') t2 on t1.no_arsip=t2.no_arsip)");
    $query = $this->db->get();
    return $query->row();
} //end proses

```

Gambar 9. Baris Program `getJumlahLokerKosong()` Pada Model (Sumber: Hasil Analisis, 2014)

Untuk mendapatkan jumlah loker kosong sistem mengambilnya pada entitas `data_loker` yang ada pada *database*. Setelah diperoleh jumlah loker kosong maka metode *hashing* dapat dijalankan yaitu dengan rumus metode seperti pada Gambar 9.

NO FILE MOD PANJANG TABEL + 1 (1)

Rumus 1. Rumus Metode Hashing
(Sumber: Hasil Analisis, 2014)

Pada Rumus 1 merupakan rumus dari metode *hashing* yang akan digunakan pada Sistem Informasi Pemberkasan Arsip Debitur dalam menentukan nomor loker untuk tempat penyimpanan arsip debitur yang melakukan proses peminjaman pada perusahaan. Penentuan loker sebagai tempat penyimpanan arsip ditentukan dari hasil modulus jumlah loker yang kosong dengan banyaknya arsip yang akan disimpan pada loker, sehingga nantinya akan diketahui nomor loker untuk tempat menyimpan arsip debitur yang melakukan peminjaman dana. Peneliti mengambil contoh dengan memasukkan 3 arsip peminjaman agunan ke dalam loker dan jumlah loker yang kosong yaitu 15 loker untuk loker jenis pinjaman agunan. Implementasi dari metode *hashing* yaitu :

$$(1 \text{ mod } 15) + 1 = 1 + 1 = 2 ; \text{ simpan 1 dilokasi 2}$$

$$(3 \text{ mod } 15) + 1 = 3 + 1 = 4 ; \text{ simpan 3 dilokasi 4}$$

$$(5 \text{ mod } 15) + 1 = 5 + 1 = 6 ; \text{ simpan 5 dilokasi 6}$$

Dari perhitungan metode *hashing* yang telah dijalankan maka akan didapat nomor loker sebagai tempat menyimpan arsip debitur seperti pada Gambar 10 merupakan hasil dari perhitungan metode *hashing* yang dilakukan berdasarkan contoh yang diberikan.

No Urut	No Loker
1	1.1.1
2	1.1.2
3	1.1.3
4	1.1.4
5	1.1.5
6	1.2.1
7	1.2.2
8	1.2.3
9	1.2.4
10	1.2.5
11	1.3.1
12	1.3.2
13	1.3.3
14	1.3.4
15	1.3.5

→ Untuk nomor arsip 1
 → Untuk nomor arsip 3
 → Untuk nomor arsip 5

Gambar 10 Hasil Dari Perhitungan Metode Hashing (Sumber: Hasil Analisis, 2014)

Setelah nomor loker setiap arsip telah diperoleh maka sistem akan menampilkannya pada tabel transaksi peminjaman yang ada pada Sistem Informasi Pemberkasan Arsip Debitur. Sistem akan menampilkan isi dari tabel transaksi peminjaman dengan memanggil function `info_pinjaman()` yang ada pada `controller`. Ketika sistem menjalankan `controller` pada `function info_pinjaman()` maka `function info_pinjaman()` akan memanggil `query` pada `class model` yaitu `function inputTabel()`. `Query` ini akan memanggil data yang ada pada tabel transaksi peminjaman dengan mengambilnya pada `database` yang ada pada entitas transaksi peminjaman. `Query function inputTabel()` dapat dilihat pada Gambar 11.

```
function inputTabel() {
  $query=$this->db->query(" select * from transaksi_peminjaman left join
  (SELECT no_arsip, status,no_loker, count(*) FROM 'loker' group by no_arsip)
  a on a.no_arsip = transaksi_peminjaman.no_arsip
  join nasabah on transaksi_peminjaman.id_nasabah=nasabah.id_nasabah
  join jaminan on transaksi_peminjaman.id_jaminan=jaminan.id_jaminan
  join jenis_pinjaman on transaksi_peminjaman.id_pinjaman =
  jenis_pinjaman.id_pinjaman
  order by transaksi_peminjaman.no_arsip desc
  ");
  return $query->result(); }

```

Gambar 11 Baris Program `getJumlahLokerKosong()` Pada Model (Sumber: Hasil Analisis, 2014)

Hasil dari `query inputTabel()` akan ditampilkan pada menu transaksi peminjaman yaitu dalam tabel transaksi peminjaman. Adapun tampilan sistem pada menu transaksi peminjaman yaitu pada Gambar 12 berikut.

Gambar 12. Menu Transaksi Peminjaman Pada Sistem Informasi Pemberkasan Arsip Debitur (Sumber: Hasil Analisis, 2014)

Nomor loker yang ditampilkan pada tabel transaksi peminjaman merupakan hasil dari perhitungan dengan menggunakan metode *hashing* untuk mendapatkan nomor loker sebagai tempat penyimpanan arsip debitur. Pada fitur ini admin dapat melakukan proses pencarian data arsip debitur, proses pergantian status debitur yang telah melunasi pinjamannya

terhadap perusahaan, dan proses peminjaman top up oleh debitur. Sedangkan untuk *user* sebagai pegawai hanya dapat melakukan proses pencarian data debitur yang ingin mereka cari. Ketika *user* melakukan klik nama debitur yang ada pada tabel transaksi maka sistem akan menampilkan tabel detail pinjaman debitur seperti terlihat pada Gambar 13.

No KTP	9090
No Arsip:	13
No Loker:	1.3.4
Nama Debitur:	Amel
Pekerjaan:	Swasta
Alamat:	Jl. Jawa
Kota Asal:	Banyuwangi
Jenis Pinjaman:	agunan
Jumlah Pencairan:	543210000
Cicilan:	4
Sisa Cicilan:	3
Tanggal Masuk:	2014-05-09
Tanggal Jatuh Tempo:	2014-09-09
Keterangan:	Modal
Status:	Masuk
No Surat Jaminan:	4577
No Surat Perjanjian:	88/88/99
Berkas Pinjaman:	AplikasiPinjaman.pdf SuratJaminan.pdf SuratPerjanjian.pdf

LUNAS OK TOP UP

Gambar 13. Tampilan Tabel Detil Pinjaman (Sumber: Hasil Analisis, 2014)

Dalam menampilkan tabel detail pinjaman sistem menjalankan kode program `detilPinjaman()` yang ada pada `controller` dan `controller` akan memanggil `class model` `getDetailPinjaman()`. Hak akses antara admin dan pegawai pada tampilan ini berbeda yaitu ketika *User* sebagai admin maka ia dapat mengoperasikan tombol "LUNAS" dan "TOP UP", dimana ketika admin menekan tombol "LUNAS" status debitur akan berubah menjadi lunas dan berkas debitur akan keluar dari direktori dimana berkas itu tersimpan. Ketika admin menekan tombol "LUNAS" pada halaman detail pinjaman maka sistem akan menjalankan `query` yang ada pada `class model` `insertStatusLokerKeluar()`. Sedangkan ketika admin mengoperasikan tombol "TOP UP" maka admin akan melakukan proses TOP UP untuk debitur yang bersangkutan, dimana sistem akan menampilkan `form` Top Up seperti pada Gambar 14 berikut.

Gambar 14 Tampilan `Form` Top Up (Sumber: Hasil Analisis, 2014)

Ketika debitur akan melakukan proses peminjaman Top Up maka admin juga melakukan pengisian data untuk peminjaman Top Up dengan menekan tombol Top Up yang ada pada halaman detail pinjaman dan sistem akan menjalankan `function Topup()` yang ada pada `controller`. Setelah sistem

mngoperasikan *controller* top up() maka sistem akan menampilkan *form* Top Up untuk memasukkan data peminjaman Top Up oleh debitur yang bersangkutan. Ketika semua *field* pada *form* Top Up terisi dan admin menekan tombol "OK" maka data debitur akan diperbaharui sesuai dengan data pinjaman baru yaitu pinjaman Top Up. Adapun kode program *controller* untuk proses peminjaman Top Up yaitu pada Gambar 15. Sedangkan untuk query pada proses penyimpanan data peminjaman Top Up terdapat pada Gambar 16 berikut.

```
public function topUp()
{
    $data_awal['ddlPinjaman'] = $this->input_debitur->ddlPinjaman();
    $data_awal['content'] = 'top';
    $data_awal['ddlNasabah'] = $this->input_debitur->ddlNasabah();

    $this->load->view('template', $data_awal);
}
```

Gambar 15 Kode Program topUp() Pada Controller (Sumber: Hasil Analisis, 2014)

```
function inputTopup($no_arsip){
    $dataTop = array(
        'no_arsip' => $no_arsip,
        'cairTopup' => $this->input->post('jumlahCair'),
        'cicilanTop' => $this->input->post('jumlahCicilan'),
        'tanggalTop' => date('Y-m-d'),
        'tempoTop' => $this->input->post('tempoTop')
    );
    $this->db->insert('topup', $dataTop);
    return $this->db->insert_id();
}
```

Gambar 16 Kode Program inputTopup() Pada Model (Sumber: Hasil Analisis, 2014)

Query inputTopup() yang digunakan pada class model berfungsi untuk menyimpan data top up debitur dan nantinya akan ditampilkan pada tabel detil pinjaman. Pada Sistem Informasi Pemberkasan Arsip Debitur juga memberikan fitur untuk mengelola data loker yang telah terpakai ataupun yang masih kosong sehingga dapat membantu admin untuk melihat loker yang masih terisi dan juga loker yang telah kosong. Untuk menampilkan halaman data loker sistem menjalankan functio dataLoker() yang ada pada *controller*. Pada Gambar 17 merupakan tampilan dari menu data loker yang ada pada sistem.

Noor Loker	No Arsip	Jenis Pinjaman	Tanggal
1.11	Kosong	Kosong	Kosong
1.12	16	agunan	2014-06-14
1.13	Kosong	Kosong	Kosong
1.14	18	agunan	2014-06-15
1.15	Kosong	Kosong	Kosong
1.21	Kosong	Kosong	Kosong
1.22	Kosong	Kosong	Kosong
1.23	Kosong	Kosong	Kosong
1.24	Kosong	Kosong	Kosong
1.25	9	agunan	2014-06-06
1.25	24	agunan	2014-06-15
1.31	Kosong	Kosong	Kosong
1.32	11	agunan	2014-06-08

Gambar 17 Tampilan Tabel Data Loker Pada Menu Data Loker (Sumber: Hasil Analisis, 2014)

Kesimpulan dan Saran

Kesimpulan

Berdasarkan analisis yang peneliti lakukan, dapat diambil beberapa kesimpulan diantaranya yaitu :

1. Metode *hashing* dapat digunakan untuk menyimpan dan mencari data dengan cepat dan efisien karena algoritma yang dibutuhkan setidaknya dapat melakukan pencarian dalam waktu yang cepat yaitu O(1) yang berarti cukup satu operasi saja untuk menemukan data yang dicari.
2. Hasil dari perhitungan metode *hashing* sebagai penentu alamat loker yang akan diisi dengan berkas debitur yang telah lolos survei peminjaman dan akan disimpan oleh perusahaan.

Saran

Beberapa saran masukan berikut diharapkan dapat memberikan perbaikan sistem dalam penelitian selanjutnya:

1. Diperlukan tampilan sistem yang lebih menarik dan tambahan fitur menu untuk hak akses *User* sebagai pegawai.
2. Diperlukan adanya pengembangan sistem dengan menggunakan ruang lingkup yang lebih luas dan diinstansi lain tidak terbatas pada perusahaan perbankan.
3. Diperlukan pengembangan sistem yang dapat mengintegrasikan sistem pemberkasan arsip debitur dengan sistem pembayaran pinjaman yang ada di perusahaan.

UcapanTerimaKasih

Penulis mengucapkan terima kasih kepada Prof. Drs Slamir, M.Comp.Sc., Ph.D selaku Ketua Program Studi Sistem Informasi Universitas Jember, Dwiretno Istiyadi S, S.T., M.Kom sebagai Dosen Pembimbing Utama dan Windy Eka Yulia Retnani S.T., M.Kom sebagai Dosen Pembimbing Anggota serta seluruh dosen Program Studi Sistem Informasi Universitas Jember. Ucapan serupa juga penulis sampaikan kepada responden yang telah memberikan kemudahan dalam pengumpulan data sehingga paper ini dapat diselesaikan.

DaftarPustaka

- [1] Amsyah, Zulkifli. (2005). *Manajemen Kearsipan*. Jakarta : PT Gramedia Pustaka Utama
- [2] Djumhana, Muhamad (2006). *Hukum Perbankan di Indonesia*. Bandung : PT Citra Aditya Bakti
- [3] Isnanto, Rizal. Adian. Nardho. *Jurnal Implementasi Metode Hash Dalam Pencarian Data Kamus Kebidanan*. Halaman 2
- [4] Ramadhan, Arif, S.Kom. (2006). *Pemrograman WEB*. Jakarta : PT Elex Media Komputindo
- [5] Sudirman. (2012). *Step by Step Desain Proyek Menggunakan UML*. Yogyakarta : CV. Andi Offset
- [6] Sugiarto, Agus dkk. (2005). *Manajemen Kearsipan Modern*. Yogyakarta : Gava Media.