PENGEMBANGAN SOFTWARE UNTUK MENGATUR PENDAPATAN SERTA PENGELUARAN UANG

Lisana
Universitas Surabaya
lisana@ubaya.ac.id

ABSTRACT

People use money everyday. Most people face some difficulties in managing their fund. People have to maintain their transactions so the money that they get (income) should be greater that the money that they spend (expense). As the result, they can have some savings that can be invested somewhere. The purpose of developing the software is to help users controlling their income and expense. Moreover, by using this software, users can monitor their payments so that they can avoid penalty risks. User can also predict how much money they should pay for the government annual tax.
Keywords: *Money Management, Money Software.*

1. Pendahuluan

Setiap hari setiap manusia selalu melakukan transaksi yang berhubungan dengan uang. Beberapa contoh transaksi yang dilakukan sehari-hari adalah pembelian bahan makanan, pembayaran pendidikan tiap bulan, pembayaran hutang, investasi berupa saham hingga pembayaran kartu kredit. Manusia juga harus berpikir bagaimana mengatur keuangan dengan baik agar bisa memenuhi kebutuhan, keinginan, memiliki simpanan untuk berjaga-jaga. Selain itu juga untuk melakukan investasi agar uang dapat bertambah sehingga dapat memenuhi impian yang sudah lama dicita-citakan. Penggunaan uang juga harus diatur pula agar dapat digunakan sesuai dengan tujuan yang sudah direncanakan. Seringkali orang tidak punya waktu untuk memikirkan hal-hal yang sangat banyak tersebut sehingga pada akhirnya sulit mengontrol penggunaan uang yang sudah didapat tiap bulan. Padahal uang yang keluar dan masuk haruslah selalu seimbang agar dana yang ada dapat dialokasikan dengan bijaksana dan terencana.

Oleh karena itu dibutuhkan sebuah software yang dapat membantu user dalam mengawasi pendapatan dan pengeluaran uang tiap bulannya, mengetahui pendapatan dari instrumen investasi yang dimiliki, dapat dengan mudah menjadwalkan pembayaran-pembayaran sehingga dapat mengurangi resiko keterlambatan pembayaran, dapat mengetahui besar pajak yang harus dibayar tiap tahunnya yang berasal dari instrumen investasi berupa deposito, emas, reksadana dan properti serta pajak yang berasal dari rekapitulasi pendapatan selama setahun. Software yang dibutuhkan haru bisa mengatur perputaran uang (*cashflow*) yang terbagi dalam enam kategori besar yaitu pendapatan (*income*), kebutuhan (*needs*), keinginan (*wants*), hutang (*debts*), tabungan (*savings*), dan investasi (*investments*) beserta pajak yang harus dibayar.

2. Analisis Kebutuhan

Pembagian kategori pendapatan dan pengeluaran dapat dikelompokkan ke dalam enam kategori utama yaitu: *Income* (pendapatan), *Needs* (kebutuhan), *Wants* (keinginan), *Debts* (hutang), *Savings* (simpanan/tabungan), dan *Investments* (investasi). Kategori utama tersebut mempunyai sub kategori agar user dapat dengan mudah mengelompokkan detail pengeluarannya. Sub kategori tersebut bersifat *expandable* (dapat ditambah) sesuai keinginan user.

Kategori-kategori atau akun-akun yang sudah disediakan oleh software dan biasanya sering digunakan oleh user dalam kehidupan sehari-hari adalah sebagai berikut:

- Pendapatan (*income*): gaji, bunga deposito, pendapatan sewa rumah, bonus, pendapatan reksadana, pendapatan lain-lain.
- Pengeluaran (*expenses*): biaya pendidikan, biaya telepon, biaya listrik, biaya air, biaya makan, biaya administrasi kartu kredit, biaya servis kendaraan, biaya beli pakaian, biaya lain-lain.

Kategori *needs* (kebutuhan) dan *wants* (keinginan) mempunyai proses-proses transaksi sebagai berikut:

- Penjadwalan pembayaran pada tiap transaksi sehari-hari. Transaksi yang dilakukan pada saat akan datang ataupun transaksi yang dilakukan berulang kali pada tiap bulannya dapat dijadwalkan sesuai dengan keinginan user. Contoh: pembayaran tagihan listrik, telepon dan air dapat dijadwalkan setiap bulan karena tanggal pembayaran tagihan tersebut rutin untuk setiap bulan.
- Transaksi pembelian secara tunai. Proses ini meliputi input sub kategori *needs* atau *wants*, memilih kategori *income* (pendapatan) atau *savings* (tabungan) yang akan digunakan untuk membayar transaksi, penerima pembayaran, keterangan dari transaksi, jumlah total yang harus dibayar dalam suatu transaksi, dan tanggal transaksi.
- Transaksi pembelian menggunakan kartu kredit. Dalam proses pembelian menggunakan kartu kredit akan masuk ke dalam tagihan kartu kredit pada bulan tertentu. Jika user terlambat melakukan pembayaran kartu kredit, maka denda dari kartu kredit tersebut merupakan biaya lain-lain yang harus diinputkan oleh user secara manual.

Kategori *debts* (hutang) meliputi perhitungan hutang sebagai berikut:

- Hutang ke lembaga keuangan. Perhitungan hutang pada lembaga keuangan user harus menginputkan besar hutang tertanggung, batas waktu pembayaran hutang, persentase bunga hutang yang harus dibayar dalam jangka waktu peminjaman.
- Hutang kartu kredit. Perhitungan hutang kartu kredit berdasarkan pada kebijakan bank yang menerbitkan kartu kredit tersebut. Input yang harus dilakukan oleh user adalah nama bank penerbit kartu kredit, limit pemakaian pembelanjaan yang dapat dilakukan oleh kartu kredit tersebut dalam satu bulan dan batas akhir pembayaran kartu kredit tersebut tiap bulannya dan biaya administrasi per tahunnya. Jika dalam transaksi kategori *needs* dan *wants* tidak ada yang menggunakan *account* kartu kredit, maka pada tanggal jatuh tempo tiap bulannya akan diperhitungkan pembayaran sebagai administrasi.
- Hutang perorangan. Pada jenis hutang perorangan, user dapat memilih untuk menginput bunga hutang atau tidak. Sifat bunga yang ada pada hutang perorangan adalah *fixed interest* (bunga tetap). Akan tetapi user diminta untuk memberikan batas akhir pembayaran hutang perorangan tersebut sehingga software dapat memberikan pesan jika tanggal jatuh tempo pembayaran sudah dilewati.

Kategori *savings* (simpanan/tabungan) yang terdiri dari tabungan, dan asuransi meliputi proses sebagai berikut:

- Input jenis *savings* berbentuk tabungan. User diminta untuk memberikan sub kategori *savings* berupa nama institusi tempat menabung, bunga rata-rata per tahun pada institusi tersebut yang bersifat dinamis.
- Transaksi penyimpanan uang (Menabung). User dapat melakukan transaksi menabung untuk menambah jumlah total dari suatu *saving account* kemudian software akan menghitung secara otomatis berapa *savings* yang ada dalam suatu *saving account* pada saat ini setelah ditambah dengan bunga tiap bulannya.
- Transaksi penarikan tabungan untuk pembayaran kategori utama *needs* (kebutuhan)*, wants* (keinginan)*,* atau *debts* (hutang). Account yang ada pada kategori *savings* juga dapat digunakan untuk transaksi kategori *needs, wants,* dan *debts*. Transaksi penarikan ini akan mempengaruhi perhitungan bunga dalam suatu *saving account* tersebut.
- Input *account* asuransi dan data-data yang dibutuhkan untuk asuransi. Pertama kali user membuat *account* premi asuransi yang tersedia, kemudian user memasukkan besar premi pertanggungan dan persentase biaya premi yang harus dibayar tiap bulan atau tiap tahunnya. Untuk asuransi kendaraan bermotor ada dua macam pilihan asuransi yaitu *total lost only* (TLO) untuk kendaraan roda dua dan roda empat dan *all risk* yang hanya untuk kendaraan roda empat. Untuk asuransi pendidikan, user diminta menginputkan usia anak yang menempuh pendidikan.
- Penjadwalan pembayaran premi asuransi. Pada setiap *account* asuransi, user dapat menjadwalkan pembayaran premi asuransi dengan menginputkan cara pembayaran menggunakan tunai atau menggunakan *saving account* yang dimiliki user dengan memberikan pesan konfirmasi terlebih dahulu untuk transaksi pembayaran yang terjadwal.

Kategori *investments* yang berupa jual beli properti, deposito, dan reksadana meliputi proses sebagai berikut:

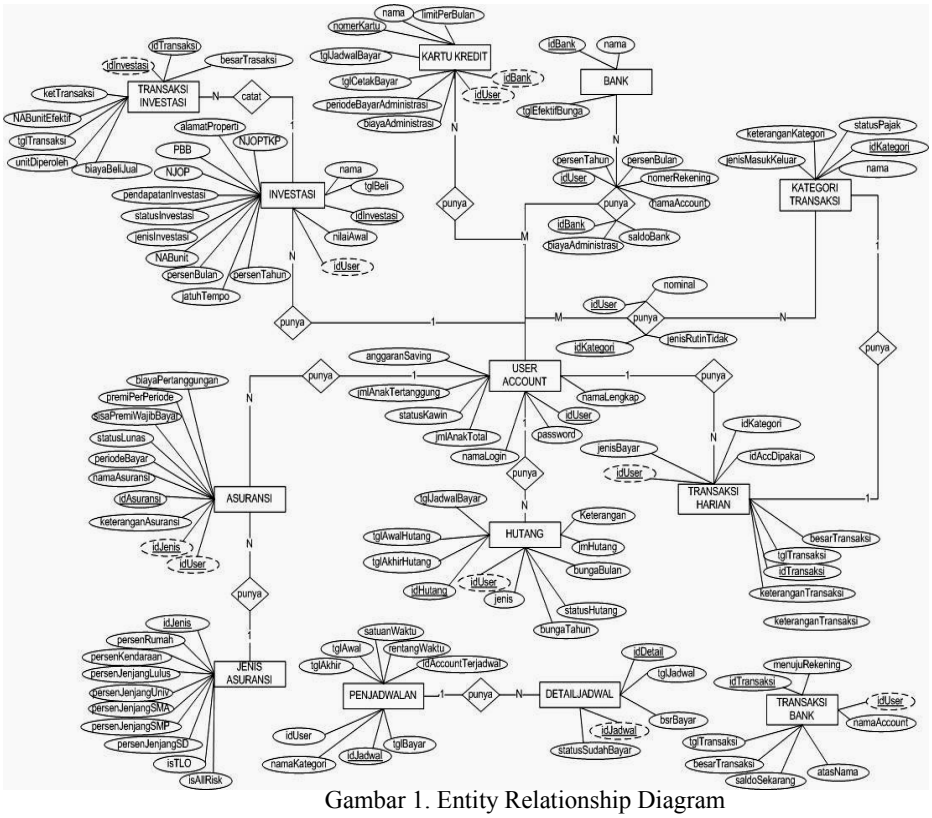
- Input *account* properti yang dimiliki. User menginputkan harga beli investasi yang dimiliki, perkiraan persentase inflasi per tahun, jenis pendapatan investasi properti berupa sewa yang akan diterima oleh user tiap periode waktu sewa.
- Transaksi penjualan properti. Pada proses transaksi penjualan, user diminta untuk menginputkan harga jual properti pada tanggal tertentu, kemudian software akan menghitung persentase laba yang diperoleh dari hasil invetasi tersebut. Dari nilai transaksi ini akan terpotong sebesar 5% dari total nilai transaksi untuk pembayaran pajak penghasilan.
- Transaksi sewa properti. Jika instrumen investasi properti yang dimiliki oleh user menghasilkan pendapatan sewa tiap periode waktu tertentu, maka user diminta untuk menginputkan besar pendapatan sewa yang didapat tiap periode pembayaran.
- Perhitungan pajak properti tiap tahun. Jika user memiliki investasi berupa properti, maka ia harus membayar pajak bumi dan bangunan (PBB) tiap tahunnya sebesar 0,5% dari Nilai Jual Kena Pajak (NJKP).
- Input jenis *investments* berbentuk deposito. User diminta untuk memberikan sub kategori *investments* berupa nama institusi tempat deposito disimpan, besar bunga per periode penyimpanan deposito yang terdiri dari pilihan satu bulan, tiga bulan, enam bulan dan satu tahun dengan sifat bunga tetap (*fixed interest*) dan minimum nominal yang diperbolehkan untuk membuka suatu *account* deposito pada institusi yang akan menerbitkan deposito.
- Transaksi perpanjangan deposito atau perubahan deposito menjadi *income* pada saat deposito tersebut jatuh tempo. Setelah user menginput *savings* berbentuk deposito dan tanggal jatuh tempo deposito tersebut sudah berakhir. Pada saat jatuh tempo, maka software secara otomatis akan memberikan pesan apakah akan melanjutkan simpanan dalam bentuk deposito tersebut atau tidak. Jika user memilih melanjutkan deposito, maka user akan kembali diminta untuk menginputkan total uang yang akan didepositokan pada institusi keuangan tersebut dan besar bunga deposito per periode penyimpanan. Jika user memilih untuk menutup deposito, maka proses transaksi akan secara otomatis account deposito tersebut berstatus tidak aktif dan tidak dapat digunakan kembali.
- Transaksi penjualan pada reksadana yang dimiliki. User diminta menginputkan data NAB/Unit pada tanggal transaksi penjualan sehingga dapat diketahui pendapatan dari hasil penjualan portofolio reksadana tersebut.
- Transaksi pembelian pada reksadana yang dimiliki. User diminta menginputkan data NAB/Unit pada tanggal transaksi pembelian sehingga dapat diketahui nominal dari pembelian portofolio reksadana tersebut.

3. Perancangan Software

Berdasarkan analisis kebutuhan yang telah didapat, maka dibuatlah rancangan data, proses, serta user interface dari software yang dibuat.

3.1 Perancangan Data

Perancangan data dilakukan dengan menggunakan *Entity Relationship Diagram* (ERD) yang kemudian dilanjutkan dengan pemetaan (*mapping*) ERD yang telah dibuat ke dalam bentuk tabel. Rancangan ERD dapat terlihat pada Gambar 1.

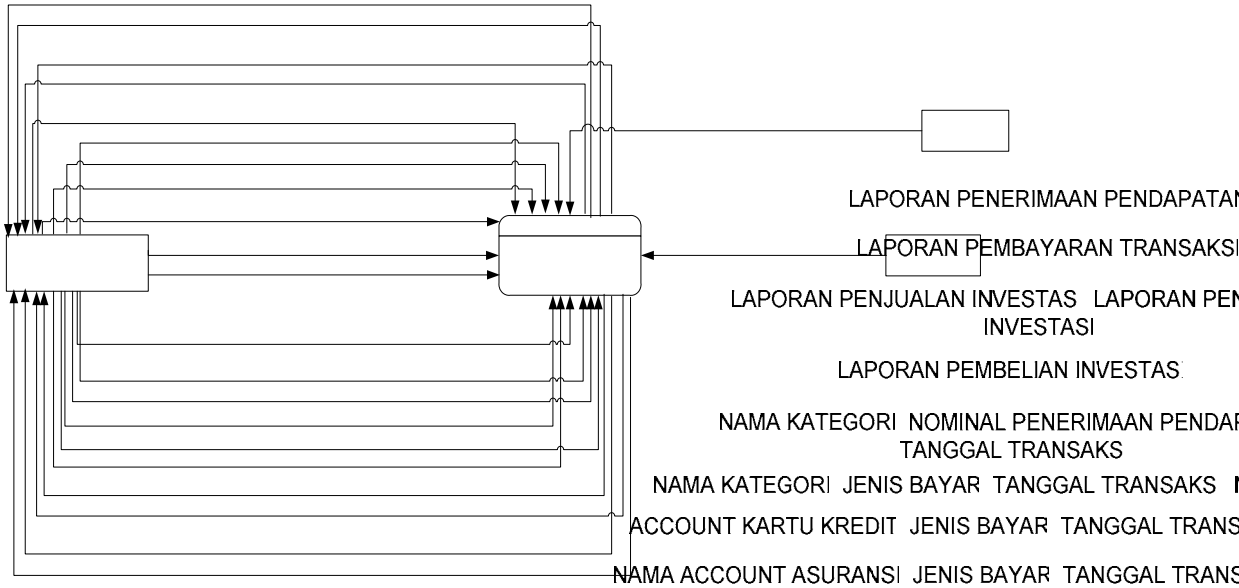


Gambar 1. Entity Relationship Diagram

3.2 Rancangan Proses

Rancangan proses dilakukan dengan menggunakan *Data Flow Diagram* (DFD). DFD adalah representasi grafik dari suatu sistem. Desain DFD untuk menggambarkan komponen-komponen suatu sistem, aliran data di antara komponen, asal, tujuan, dan penyimpanan dari data tersebut. DFD Level 0 dapat terlihat pada Gambar 2. DFD level 0 terdiri dari satu proses yaitu sistem informasi manajemen keuangan pribadi, dan tiga entiti yang berperan dalam proses sistem informasi manajemen keuangan pribadi yaitu user, perusahaan asuransi dan bank. Entiti perusahaan asuransi dan bank hanya memberikan data-data master yang dibutuhkan dalam pembayaran transaksi-transaksi yang ada pada sistem informasi manajemen keuangan pribadi.

Proses sistem informasi manajemen keuangan pribadi menerima input data antara lain data account investasi, data account asuransi, data pribadi, data rekening bank, data kartu kredit yang dimiliki user dan data hutang. Sedangkan data yang menjadi output dari proses ini antara lain tampilan data asuransi, tampilan data investasi, tampilan data hutang, laporan pembayaran transaksi hutang maupun transaksi harian, laporan transaksi jual beli yang terjadi pada account investasi dan laporan penerimaan pendapatan.



Gambar 2. Data Flow Diagram Level 0

3.3 Perancangan User Interface

Sebelum sistem dibuat, terlebih dahulu dibuat suatu desain user interface (*user interface design*). Rancangan user interface yang dibuat antara lain:

- Form Login dan Daftar User Baru. Untuk dapat menjalankan program software maka user harus menginputkan *User name* dan *password*.
- Menu Utama. Menu utama berisi beberapa menu yang terdiri dari data user, ganti user, tambah kategori, skenario prediksi, navigasi TOKEN HOME, navigasi transaksi harian, navigasi tabungan, navigasi asuransi, navigasi hutang, navigasi investasi, dan laporan.
- Form Navigasi TOKEN HOME. Form yang memberikan informasi tentang pendapatan dan pengeluaran yang ada dalam bulan tertentu, account kartu kredit, account tabungan, jadwal transaksi yang dimiliki oleh user dengan hanya klik tabulasi jadwal transaksi.
- Form Navigasi Transaksi. Form ini berfungsi untuk menginputkan data transaksi harian, menjadwalkan kategori transaksi yang dimiliki oleh user, memilih account yang akan digunakan untuk transaksi pembayaran, account kartu kredit, atau tabungan.
- Form Navigasi Hutang, Asuransi dan Investasi. Form navigasi hutang digunakan untuk melihat, memasukkan, dan mengubah data hutang yang dimiliki.
- Form Navigasi Tabungan. Form navigasi tabungan digunakan untuk memasukkan transaksi yang menggunakan suatu account tabungan dan ada transaksi menabung ataupun untuk transfer ke rekening lain.
- Form Master Kategori Transaksi. Form ini berfungsi untuk menambah dan mengubah kategori transaksi. Jika transaksi bersifat rutin maka user dapat menginputkan data periode jadwal, tanggal awal, dan batas jadwal.
- Form Master Account Bank. Form ini untuk menambah rekening yang dimiliki oleh user dalam suatu bank tertentu.
- Form Master Account Kartu Kredit. Form ini untuk menambah account kartu kredit yang dimiliki oleh user. Field tanggal cetak adalah field untuk menunjukkan tanggal berapa dalam satu bulan untuk batas transaksi menggunakan kartu kredit agar transaksi tersebut masuk dalam bulan yang sama. Field tanggal jatuh tempo adalah tanggal dimana user harus membayar tagihan agar tidak dikenakan denda kartu kredit.
- Form Master Account Asuransi. Form ini untuk menambah account asuransi yang dimiliki oleh user. Software hanya menyediakan tiga jenis asuransi yaitu asuransi pendidikan, asuransi kendaraan dan asuransi rumah dimana user dapat menginputkan persentase pembayaran untuk tiap periode pembayaran.
- Form Master Account Investasi. Form master account investasi mempunyai empat sub investasi yang dapat dipilih oleh user, yaitu investasi berupa deposito dimana user harus menginputkan field tanggal setoran awal, tanggal jatuh tempo, dan bunga deposito.
- Form Master Account Hutang. Form master account hutang berfungsi untuk menginputkan data hutang yang dimiliki oleh user. Software menyediakan dua pilihan jenis hutang yaitu hutang pribadi atau perorangan dan hutang ke lembaga keuangan, misalnya hutang bank atau hutang ke koperasi simpan pinjam.
- Form Transaksi Pembayaran Hutang. Form transaksi pembayaran hutang digunakan untuk pembayaran account hutang yang dimiliki beserta daftar jadwal pembayaran.

- Form Transaksi Investasi. Form transaksi investasi terdiri dari empat pilihan transaksi yaitu transaksi penjualan reksadana, transaksi penjualan emas, transaksi penjualan properti dan transaksi perpanjangan deposito. Untuk transaksi penjualan reksadana, user diminta untuk menginputkan jumlah unit yang dijual, nilai NAB per unit jual dan subscription fee sebagai tambahan biaya untuk transaksi penjualan.

4. Implementasi dan Uji Coba

Setelah rancangan dibuat kemudian dilakukan implementasi terhadap rancangan tersebut. Untuk implementasi dari rancangan basis data digunakan Microsoft Access. Untuk implementasi dari rancangan proses digunakan bahasa pemrograman Microsoft Visual Basic .NET 2005 dengan .NET Framework 2.0. Sedangkan implementasi dari rancangan keseluruhan laporan digunakan Crystal Report 10. Semua rancangan yang telah dibuat telah dapat diimplementasikan dengan baik.

Software yang dihasilkan kemudian diujicoba untuk meyakinkan bahwa software telah bebas dari kesalahan (*errors*). Uji coba dilakukan dengan menjalankan semua fasilitas yang terdapat pada software yang dihasilkan tersebut. Pada awal software dijalankan akan muncul form login yang bertujuan untuk melindungi data transaksi pribadi user. Sebelum user dapat menggunakan software, user harus melakukan registrasi pembuatan account. Uji coba dimulai dari proses pembuatan account user baru yang dilanjutkan dengan melakukan proses pembuatan account rekening dan kartu kredit. Setelah itu dilakukan proses pembuatan kategori baru beserta penjadwalannya. Langkah berikutnya adalah pengujian proses pembuatan account hutang yang dimiliki beserta dengan jadwal pembayaran. Dilanjutkan dengan pengujian beberapa proses yang lain yaitu: proses pembuatan account asuransi, proses pembuatan account investasi, proses pembayaran kategori terjadwal, proses input transaksi harian, dan proses transaksi penjualan investasi. Berbagai kombinasi data diinputkan ke software untuk meyakinkan bahwa software telah benar-benar bebas dari kesalahan.

5. Kesimpulan

Dari hasil uji coba yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Semua proses yang berhubungan dengan pendapatan dan pengeluaran uang telah dapat berjalan dengan baik dan pembuatan berbagai macam laporan dapat dilakukan dengan lebih mudah, efisien, dan terstruktur.
2. Fasilitas penjadwalan yang tersedia dapat membantu user mengetahui transaksi yang telah jatuh tempo dan dapat dengan segera melakukan transaksi pembayaran.
3. Adanya fasilitas pembuatan laporan pajak pribadi dapat dilakukan dengan mudah dan efisien.

Daftar Pustaka

[1] Elmasri, R.; Navathe S.B. (2000). *Fundamentals of Database Systems (2nd ed.)*. The Benjamin/Cummings Publishing Company, Inc., Redwood City.
[2] Aditya Eka N. (2009). *Pembuatan Aplikasi Sistem Informasi Manajemen Keuangan Pribadi*.
[3] Manulang, B.E., dkk. (2003). *Soal Jawab Perpajakan (Kunci Sukses Ujian Brevet Pajak)*. Citramedia, Sidoarjo.
[4] Pujawan, I Ny. (2004). *Ekonomi Teknik*. Guna Widya, Surabaya.