

## APLIKASI PENGELOLAAN ZAKAT, INFAQ DAN SHODAQOH (ZIS) BERBASIS CLIENT SERVER DI MASJID AL-MUHAJJIRIN BANDUNG BARAT

Ridwan Nurlianto<sup>1</sup>, Supriady,<sup>2</sup> Woro Isti Rahayu<sup>3</sup>

Jurusan Teknik Informatika

Politeknik Pos Indonesia

la\_2503@ymail.com, goku\_sakti2005@yahoo.com, wistirahayu@yahoo.com

### ABSTRAK

Pengelolaan merupakan suatu proses pendataan mengenai banyaknya data-data yang harus tersusun secara rapih dan terorganisir. Dalam menangani suatu transaksi amal zakat, infaq maupun shodaqoh, pihak Pengelola melakukan beberapa proses mulai dari pencatatan tentang data donator, data penerima, input dana serta sampai kepada penyerahan dana yang akan diberikan hingga pembuatan laporan-laporan. Dengan banyaknya proses transaksi yang dilakukan, suatu pengelola membutuhkan sebuah aplikasi yang dapat mempermudah pengelolaan data, khususnya untuk data pengelolaan zakat, infaq dan shodaqoh. Oleh karena itu, dibuatlah aplikasi "Pengelolaan Zakat, Infaq dan Shodaqoh".

Aplikasi ini dibuat dengan menggunakan program aplikasi *Visual Basic.Net 2005*, *SQL Server2000*. Sistem operasi yang digunakan adalah XP SP2 yang berjalan di *client* maupun *server*.

Aplikasi ini menyediakan menu-menu dalam bentuk form pengolahan data-data Donatur, data-data Penerima, Input Dana Zakat, Infaq maupun Shodaqoh, Total Dana yang didapat secara keseluruhan dan Penyerahan kepada siapa dana yang di input akan diberikan serta pembuatan laporan.

**Kata Kunci:** *Pengelolaan Zakat, Infaq dan Shodaqoh, client, server, event, Visual Basic.Net 2005, SQL Server2000*

### I. PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi aplikasi semakin pesat dan meningkat, bahkan perusahaan-perusahaan, rumah sakit, dan bahkan lembaga kemanusiaan nasional seperti Pos Keadilan Peduli Umat (PKPU) sudah menggunakan aplikasi yang diolah sedemikian rupa agar dapat mengelola semua data yang ada, bukan hanya data tetapi juga barang atau memenejemen keuangan juga menggunakan suatu aplikasi. Namun apakah aplikasi yang berjalan sudah sesuai kebutuhan dan efisiensi?

Dengan penggunaan dan penerapan aplikasi akan meningkatkan pengalokasian baik tenaga manusia, peralatan, dana dan sumber-sumber lainnya, disamping kontrol yang baik dan

keyakinan didalam melakukan evaluasi tentang status dan progress suatu pekerjaan.

Penerapan aplikasi dalam organisasi dan manajemen secara baik akan sangat berpengaruh terhadap suksesnya misi yang ditetapkan, karena sepenuhnya dapat mengontrol jalannya organisasi dan dapat digunakan dalam meningkatkan total integritas diantara unit-unit kerja yang dibawahnya dan meningkatkan tanggung jawab bagi tiap-tiap unit kerja lainnya.

Khususnya masjid Al-Muhajirin di daerah Bandung Barat. Dalam penerapan pengelolaan zakat, infaq, shodaqoh masih menggunakan cara manual atau pendataan dalam sebuah buku yang di khawatirkan terjadi kesalahan data-data penting yang dibutuhkan. Didalam pengelolaannya banyak kendala yang dialami mulai dari pengelolaan sampai pada pembagian zakat, mengelola data-data yang ada seperti data pemberi zakat, data keuangan dan data penerima zakat. Kemudian Dalam pendataan warga setempat yang belum mampu masih sangat rumit. Harus dilakukan pendataan warga per kepala keluarga, sehingga nantinya semua dana

amal akan tepat sasaran diberikan kepada yang berhak menerimanya.

Berdasarkan uraian diatas, maka diambil judul "Aplikasi Pengelolaan Zakat, Infaq dan Shodaqoh (ZIS) Berbasis *Client Server* di Masjid Al-Muhajirin Bandung Barat" pada Tugas Akhir ini.

### 1.2 Identifikasi Masalah

Identifikasi masalah merupakan suatu tahap pengumpulan atau pendataan masalah mengenai kekurangan aplikasi yang telah ada sebelumnya, sehingga diharapkan menjadi suatu yang layak untuk dianalisis dan dilakukan perancangan aplikasi baru serta mencari apa solusi untuk memecahkan masalah-masalah tersebut melalui fungsi dari aplikasi baru yang akan bangun.

Berdasarkan latar belakang yang telah diuraikan diatas, pada kenyataan saat sekarang ini, pengelolaan masalah keuangan, data mengenai orang yang akan melakukan transaksi zakat, infaq dan shodaqoh, data penduduk setempat yang memang berhak menerimanya masih memiliki sistem yang manual atau belum tersentuh sedikitpun dengan teknologi yang sudah sangat pesat berkembang pada saat sekarang.

Adapun beberapa masalah yang terdapat dalam kehidupan sehari-hari yang bersangkutan dengan penerapan pengelolaan zakat, infaq dan shodaqoh dilingkungan masyarakat, yaitu :

1. Dalam pengelolaan datanya masih menggunakan cara manual yang di khawatirkan terjadi kesalahan data.
2. Kurangnya penertiban administrasi keuangan apabila masih menggunakan cara atau proses manualisasi dalam transaksi zakat, infaq dan shodaqoh.
3. Masih kurang efisien dalam pendataan warga kurang mampu per kepala keluarga yang memang berhak menerima.

### 1.3 Tujuan

Tujuan disini adalah membahas tentang apa tujuan pembuatan aplikasi yang akan dibangun. Dilihat dari uraian identifikasi masalah diatas, bahwa yang menjadi tujuan dibangunnya suatu aplikasi pengelolaan ini adalah sebagai berikut :

1. Merancang dan membangun suatu aplikasi pengelolaan zakat, infaq dan shodaqoh agar dalam penerapannya lebih terorganisir dan tepat sasaran.
2. Menerapkan penertiban administrasi keuangan dalam transaksi zakat, infaq dan shodaqoh.

3. Menerapkan cara baru yang lebih efisien dan lebih cepat dalam pemrosesan setiap transaksi dari sistem lama yang masih menggunakan sistem manualisasi untuk menghindari kesalahan data yang sering terjadi.

### 1.4 Ruang Lingkup

Batasan-batasan permasalahan dalam pembuatan aplikasi ini adalah sebagai berikut :

1. Pengelolaan zakat, infaq dan shodaqoh, pendataan donatur, pendataan keuangan serta pendataan masyarakat yang berhak menerima.
2. Mempermudah melakukan pengelolaan mengenai transaksi zakat, infaq dan shodaqoh secara terorganisir dan tepat sasaran dalam penerapannya.
3. Dalam perancangan aplikasi ini menggunakan bahasa pemograman Visual Basic. *Net 2005* berbasis *client-server*.

## II. LANDASAN TEORI

### 2.1 Sistem Informasi

Sistem Informasi merupakan sistem terkomputerisasi yang memiliki kelengkapan data dan terintegritas yang memanfaatkan teknologi informasi dan berupa perangkat lunak komputer atau software yang dibuat untuk melakukan tugas-tugas penanganan penangan dan pengelolaan data. Dengan sistem informasi akademik, manajemen dapat memperoleh informasi apa saja yang berkaitan dengan instansi dengan mudah, cepat, akurat dan on-line. Informasi tersebut dapat diperoleh hanya dengan beberapa ketukan di keyboard komputer (dikenal dengan istilah *information in your fingertips*).

Sistem informasi dibangun dengan siklus SDLC (*System Development Life Cycle*). Siklus SDLC akan memberi panduan dan prosedur bagi semua yang terlibat proyek pembangunan sistem informasi dengan beberapa manfaat seperti alokasi waktu yang terencana, mengurangi resiko kegagalan proyek, memastikan bahwa semua kebutuhan tercakup dalam proyek, mengidentifikasi masalah teknis dan manajerial yang mungkin muncul, mengukur kemajuan jalannya proyek dan mempermudah pengaturan sumber daya serta anggaran.

SDLC adalah proses dalam membangun sistem informasi melalui beberapa langkah. Ada beberapa model SDLC salah satunya adalah *waterfall*. Model ini disebut *waterfall* karena dikerjakan langkah per langkah seperti air mengalir. Model lain SDLC misalnya *fountain*,

*spiral, rapid prototyping, incremental, build and fix dan synchronize & stabilize.*

Dalam sebuah siklus SDLC terdapat tujuh langkah. Langkah tersebut adalah:

- 1) Perencanaan (*Planning*), yaitu membuat perencanaan yang berkaitan dengan proyek sistem informasi.
- 2) Analisa (*Analysis*), yaitu membuat analisa workflow manajemen yang sedang berjalan.
- 3) Desain (*Design*), yaitu membuat desain workflow manajemen dan desain pemrograman yang diperlukan untuk pengembangan sistem informasi.
- 4) Pengembangan (*Development*), yaitu tahap pengembangan sistem informasi dengan menulis program-program yang diperlukan.
- 5) Testing (*Testing*), yaitu melakukan pengujian terhadap sistem informasi yang telah dibuat.
- 6) Implementasi (*Implementation*), yaitu menerapkan sistem informasi yang telah dibuat untuk digunakan user.

Pengoperasian dan pemeliharaan (*operations and maintenance*), yaitu kegiatan untuk mendukung beroperasinya sistem informasi. [1]

## 2.2 Pengertian Zakat, Infaq dan Shodaqoh

### 2.2.1 Zakat

Zakat adalah sedekah yang wajib dikeluarkan umat Islam menjelang akhir bulan Ramadhan, sebagai pelengkap ibadah puasa. Zakat merupakan salah satu rukun ketiga dari Rukun Islam. Secara harfiah zakat berarti "tumbuh", "berkembang", "menyucikan", atau "membersihkan". Sedangkan secara terminologi syaria'ah, zakat merujuk pada aktivitas memberikan sebagian kekayaan dalam jumlah dan perhitungan tertentu untuk orang-orang tertentu sebagaimana ditentukan.

#### 2.2.1.1 Hukum Zakat

Zakat merupakan salah satu rukun Islam, dan menjadi salah satu unsur pokok bagi tegaknya syariat Islam. Oleh sebab itu hukum zakat adalah wajib (fardhu) atas setiap muslim yang telah memenuhi syarat-syarat tertentu. Zakat termasuk dalam kategori ibadah, seperti : shalat, haji dan puasa yang telah diatur secara rinci dan paten berdasarkan Al-Qur'an dan As Sunnah serta sekaligus merupakan amal sosial kemasyarakatan dan kemanusiaan yang dapat berkembang sesuai dengan perkembangan umat manusia. Allah SWT berfirman dalam surat Al-Baqarah ayat 83 :

أَقِيمُوا الصَّلَاةَ وَآتُوا الزَّكَاةَ وَأُولُوا... (33 : قرقبلا )

Artinya : "Dan dirikanlah shalat dan tunaikanlah zakat." (QS. Al Baqarah : 83)

خُذْ مِنْ أَمْوَالِهِمْ صَدَقَةً تُطَهِّرُهُمْ وَتُزَكِّيهِمْ بِهَا وَصَلِّ عَلَيْهِمْ إِنَّ صَلَاتَكَ

سَكَنٌ لَهُمْ وَاللَّهُ سَمِيعٌ عَلِيمٌ ﴿١٠٣﴾

Artinya : "Ambillah zakat dari sebagian harta mereka dengan zakat itu kamu membersihkan dan mensucikan mereka" (QS. At Taubah : 103).

#### 2.2.1.2 Macam-macam Zakat

Zakat terbagi atas dua tipe yakni :

##### 1. Zakat Fitrah

Zakat yang wajib dikeluarkan muslim menjelang Idul Fitri pada bulan Ramadhan. Besar Zakat ini setara dengan 2,5 kilogram makanan pokok yang ada di daerah bersangkutan.

##### 2. Zakat Maal (Harta)

Mencakup hasil perniagaan, pertanian, pertambangan, hasil laut, hasil ternak, harta temuan, emas dan perak. Masing-masing tipe memiliki perhitungannya sendiri-sendiri.

Sedangkan orang-orang yang berhak menerima zakat adalah sebagai :

1. Fakir - Mereka yang hampir tidak memiliki apa-apa sehingga tidak mampu memenuhi kebutuhan pokok hidup.
2. Miskin - Mereka yang memiliki harta namun tidak cukup untuk memenuhi kebutuhan dasar untuk hidup.
3. Amil - Mereka yang mengumpulkan dan membagikan zakat.
4. Muallaf - Mereka yang baru masuk Islam dan membutuhkan bantuan untuk menyesuaikan diri dengan keadaan barunya
5. Hamba Sahaya yang ingin memerdekakan dirinya
6. Gharimin - Mereka yang berhutang untuk kebutuhan yang halal dan tidak sanggup untuk memenuhinya
7. Fisabilillah - Mereka yang berjuang di jalan Allah (misal: dakwah, perang dsb)
8. Ibnu Sabil - Mereka yang kehabisan biaya di perjalanan.

Serta orang-orang yang tidak berhak menerima zakat adalah :

1. Orang kaya. Rasulullah bersabda, "Tidak halal mengambil sedekah (zakat) bagi orang yang kaya dan orang yang mempunyai kekuatan tenaga." (HR Bukhari).
2. Hamba sahaya, karena masih mendapat nafkah atau tanggungan dari tuannya.
3. Keturunan Rasulullah. Rasulullah bersabda, "Sesungguhnya tidak halal bagi kami (ahlul

bait) mengambil sedekah (zakat)." (HR Muslim).

4. Orang yang dalam tanggungan yang berzakat, misalnya anak dan istri.
5. Orang kafir.

### 2.2.2 Infaq

Infaq adalah mengeluarkan harta yang mencakup zakat dan non zakat. Infaq ada yang wajib dan ada yang sunnah. Infaq wajib diantaranya zakat, kafarat, nadzar, dll. Infaq sunnah diantaranya, infak kepada fakir miskin sesama muslim, infak bencana alam, infak kemanusiaan. Terkait dengan infak ini Rasulullah SAW bersabda dalam hadits yang diriwayatkan Bukhari dan Muslim ada malaikat yang senantiasa berdo'a setiap pagi dan sore : "Ya Allah SWT berilah orang yang berinfaq, gantinya. Dan berkata yang lain : "Ya Allah jadikanlah orang yang menahan infak, kehancuran".

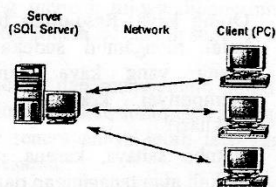
### 2.2.3 Shodaqoh

Sedekah asal kata bahasa Arab adalah shadaqoh yang berarti suatu pemberian yang diberikan oleh seorang muslim kepada orang lain secara spontan dan sukarela tanpa dibatasi oleh waktu dan jumlah tertentu. Juga berarti suatu pemberian yang diberikan oleh seseorang sebagai kebajikan yang mengharap ridho Allah SWT dan pahala semata.

Pahala sedekah akan lenyap bila si pemberi selalu menyebut-nyebut sedekah yang telah ia berikan atau menyakiti perasaan si penerima.

### 2.3 Client / Server

*Client / server* merupakan tipikal dari sebuah aplikasi *two-tier* dengan banyak *client* dan sebuah *server* yang dihubungkan melalui sebuah jaringan. Komputer *client front end* terdiri atas aplikasi yang ditulis dalam bahasa pemrograman. Sedangkan komputer *server (back end)* merupakan *Database Management System (DBMS)* [2].



Gambar 1 Arsitektur Aplikasi Two-Tier

aplikasi *client / server* yaitu :

1. *Client* melakukan sebuah permintaan ke *server*. Biasanya jaringan komputer telah terhubung dengan topologi jaringan tertentu.
2. Ketika *server* menerima permintaan layanan, *server* mengatur dan membawa permintaan tersebut kemudian mengirimkan balik sebuah pesan respon ke *client* yang meminta.
3. Respon tersebut dapat berupa sekumpulan *record*, pesan kesalahan, atau sebuah nilai *Boolean* yang mengindikasikan apakah permintaan telah dieksekusi dengan sukses.

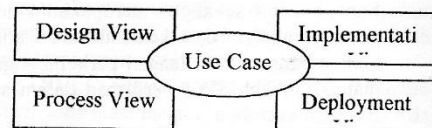
Dalam *client / server*, *client* bertanggung jawab untuk bagian dari aplikasi yang berinteraksi dengan *user*, termasuk logika bisnis dan komunikasi dengan *server database*. Sedangkan *server database* menangani manajemen data, keamanan, *query*, *trigger*, *prosedur* tersimpan dan penanganan kesalahan.

Arsitektur *client/server* dapat mengurangi beban pemrosesan dari komputer sentral ke komputer *client*, sehingga meskipun semakin banyak *user* bertambah pada aplikasi *client/server*, kinerja *server file* tidak akan menurun dengan cepat. Dengan *client / server*, *user* dari berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal.

### 2.4 UML (Unified Modelling Language)

UML (*Unified Modelling Language*) merupakan kesatuan dari bahasa pemodelan yang mengkombinasikan metode OO (*Object Oriented*) yang telah ada sebelumnya yang digunakan sebagai alat bantu untuk memvisualisasikan dan mendokumentasikan hasil analisa dan design bagi pengembang sistem yang dapat menerjemahkan diagram-diagram yang ada di dalamnya menjadi kode yang siap untuk dijalankan [3].

UML dibangun atas model 4+1 view, yaitu sebuah sistem dideskripsikan dalam 5 view yang merupakan penjelasan lengkap tentang sistem dengan menggabungkan informasi-informasi yang ada pada kelima view. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan diuji. Berikut adalah gambar model tersebut :



Gambar 2 Model 4+1 view

Konsep umum dalam model arsitektur

1. *Use case view*, mendefinisikan perilaku eksternal sistem, mempunyai peran khusus untuk mengintegrasikan *content* ke view yang lain.
2. *Design view*, berisi definisi komponen *program*, *class-class* utama serta spesifikasi datanya, perilaku dan interaksinya.
3. *Implementation View*, menjelaskan komponen-komponen fisik dari sistem yang akan dibangun.
4. *Process View*, berhubungan dengan hal-hal yang berkaitan dengan *concurrency* di dalam sistem.
5. *Deployment view*, menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik, misalnya jaringan komputer.

UML menyediakan beberapa diagram yang bisa membantu mendefinisikan suatu aplikasi. Diagram-diagram tersebut adalah sebagai berikut :

1. *Use Case Diagram*, terdiri dari sekumpulan skenario yang dilakukan oleh aktor-aktor untuk mendeskripsikan sistem yang akan dibangun.
2. *State Machine Diagram*, berfungsi untuk membantu analisis, *designer* dan *developer* dalam memahami perilaku objek dalam sistem.
3. *Sequence Diagram*, berisi sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek dalam *use case* untuk menggambarkan perilaku pada sebuah skenario.
4. *Collaboration Diagram*, bentuk lain dari *sequence* diagram yang diorganisir menurut ruang/ *space* yang memungkinkan untuk memodelkan pengiriman sebuah pesan ke banyak objek pada *class* yang sama.
5. *Activity Diagram*, digunakan untuk menunjukkan tahapan aliran kerja, pengambilan keputusan dan percabangan.
6. *Component Diagram*, merepresentasikan *component software*, berisi *component*, *interface*, dan *relationship*. *Component* mewakili potongan-potongan yang independen yang dapat diperbaharui sewaktu-waktu.
7. *Deployment Diagram*, merupakan gambaran sistem secara fisik dapat terlihat, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*.
8. *Package Diagram*, merupakan pengelompokan konstruksi atau elemen-elemen menjadi *level* yang lebih tinggi, yang menggambarkan saling ketergantungan antara *component-component* utama pada sistem.
9. *Object Diagram*, merupakan gambaran objek-objek secara ringkas dalam sistem pada suatu waktu yang berfungsi sebagai *instance diagram* yang menunjukkan *instance-instance* dari *class*.

## 2.5 Database

Database adalah sekumpulan data yang berfungsi sebagai penyedia informasi bagi pengguna. Sebelum membuat database, hal penting yang harus dilakukan adalah mendesain database. Pemodelan yang baik untuk mendesain database adalah ERD (*Entity Relationship Diagram*). Selain itu, hal yang harus diperhatikan ketika mendesain database adalah membuat relasi-relasi yang benar diantara tabel.

1. *Entity* adalah objek, tempat, orang, konsep atau aktivitas yang mudah diidentifikasi. *Entity* dilambangkan dengan kotak. Contoh: mahasiswa, mata kuliah dan dosen.
2. Atribut adalah sifat-sifat dari sebuah *entity* yang membedakan *entity* yang satu dengan *entity* yang lain. Contoh: *entity* dosen mempunyai atribut kode dosen, nama dosen, alamat.
3. *Relationship* ialah bagian yang sangat penting di dalam mendesain *database* yang merupakan penghubung antara suatu *entity* dengan *entity* lain. Sebuah relasi mungkin merupakan penghubung dengan dirinya sendiri dan mungkin juga beberapa relasi berada di antara *entity* yang sama. Sebuah *relationship* dilambangkan dengan belah ketupat.

Terdapat tiga tipe *relationship* diantaranya :

- a. *Relationship one to one* (1:1), satu *entity* hanya berelasi dengan satu *entity* lainnya.
- b. *Relationship one to many* (1:m), satu *entity* bisa mempunyai relasi lebih dari satu pada *entity* lainnya.
- c. *Relationship many to many* (m:m), banyak *entity* bisa berelasi dengan banyak *entity* lainnya.

Adapun Struktur dari data relasi adalah sebagai berikut :

- 1) *Candidate key* adalah atribut yang secara unik mengidentifikasi sebuah baris dan berfungsi sebagai calon *primary key*.
- 2) *Primary key* adalah *candidate key* yang dipilih untuk mengidentifikasi tiap baris secara unik dan tidak boleh ada nilai NULL.
- 3) *Alternate key* adalah *candidate key* yang tidak dipilih sebagai *primary key*.
- 4) *Composite key* adalah suatu kolom yang digunakan sebagai *primary key* karena pada kondisi tertentu suatu atribut tidak dapat digunakan untuk mengidentifikasi baris secara unik.
- 5) *Foreign key* adalah *field* pada tabel yang merupakan *primary key* pada tabel yang lain.

## 2.6 Microsoft SQL Server 2000

Microsoft SQL Server 2000 adalah perangkat lunak RDBMS (*Relational Database Management System*) yang mendukung aplikasi dengan arsitektur client/server (*two-tier*) dimana sebagian proses dilakukan oleh server dan sebagian lagi dilakukan oleh aplikasi sehingga dapat mengurangi lalu lintas jaringan karena SQL Server hanya memberikan data yang diperlukan saja oleh client.

Database pada SQL Server dibagi menjadi dua bagian yaitu database sistem dan database user. Database sistem adalah database yang ada pada waktu database diinstal yaitu database master, model, tempdb, pubs, northwind, dan msdb. Database user adalah database yang diciptakan oleh user sendiri.

*Transact-SQL (Structure Query Language)* adalah suatu bahasa pemrograman yang digunakan untuk mengakses database, melakukan query, update, serta mengelola hubungan sistem database [5].

Tiga jenis *Transact-SQL* yang dimiliki SQL Server yaitu :

1. *Data Definition Language (DDL)* digunakan untuk mengelola objek database seperti database, tables, dan views. Perintah-perintah yang termasuk DDL antara lain create, alter, dan drop.
2. *Data Control Language (DCL)* berfungsi untuk mengontrol hak-hak pada objek database. Perintah-perintah yang termasuk DCL yaitu grant, revoke, dan deny.
3. *Data Manipulation Language (DML)* berfungsi untuk memanipulasi data yang ada di dalam tabel. Perintah-perintah DML antara lain select, insert, update, dan delete.

Beberapa komponen-komponen dari Microsoft SQL Server 2000 antara lain :

1. Database adalah sekumpulan data yang berfungsi sebagai penyedia informasi bagi pengguna.
2. Tabel adalah tempat untuk menyimpan data dalam database.
3. View adalah tabel virtual yang berisi kolom sebuah tabel atau lebih untuk mempermudah pengguna mengakses atau melihat data yang sering dibutuhkan.
4. Store Procedure adalah kumpulan pernyataan SQL yang disimpan ke dalam database dengan suatu nama dan mendukung penggunaan variabel dari pengguna, eksekusi kondisional dan fasilitas pemrograman yang powerful.
5. Trigger adalah sejenis store procedure yang dijalankan ketika aktivitas insert, update dan delete dilakukan.

Pada SQL Server, pembuatan database beserta objek-objek di dalamnya dapat dilakukan dengan dua cara yaitu :

- a) *Query Analyzer* yaitu layar editor untuk melakukan serangkaian aktivitas yang berhubungan dengan query. Semua aktivitas yang dilakukan dapat disimpan dalam suatu file dan dapat dijadikan sebagai back up file.
- b) *Enterprise Manager* merupakan program utama dari Microsoft SQL Server yang memiliki beberapa tools dan fungsi-fungsi pokok dalam mengelola database server.

## 2.7 Visual Basic.Net 2005

Visual Basic.NET atau VB.Net merupakan bahasa pemrograman yang biasanya digunakan oleh para mahasiswa sampai praktisi programmer professional. Bahasa pemrograman ini adalah pengembangan dari software Microsoft Visual Basic sebelumnya yang dapat digunakan dalam pembuatan aplikasi di lingkungan kerja berbasis Windows.

Komponen-komponen lingkungan kerja VB.Net dan fungsinya :

1. *Toolbox*, berfungsi untuk tempat meletakkan objek-objek yang menjadi alat untuk memprogram.
2. *Solution Explorer*, berfungsi untuk melihat item-item penyusun sebuah proyek / solution. Berikut ini adalah item-item penyusun proyek :
  - ↳ Form dan kodenya, mempunyai ekstensi .vb
  - ↳ Class, mempunyai ekstensi .vb
  - ↳ Module, mempunyai ekstensi .vb
  - ↳ Class Diagram, mempunyai ekstensi .cd
  - ↳ File XML, mempunyai ekstensi .xml
  - ↳ Project File, mempunyai ekstensi .vbproj
  - ↳ File laporan Crystal Report, mempunyai ekstensi .rpt
3. *Propeties*, berfungsi untuk melihat nilai properti dari sebuah objek atau class.
4. Area Kerja, merupakan jendela untuk melakukan kegiatan pengisian kode ketika berupa tampilan kode (code view) serta tempat mengartur desain form ketika berupa tampilan desainer (designer view).
5. *Main Menu*, berfungsi sebagai tempat mengakses fungsi-fungsi di VB.Net.
6. *Toolbar*, berfungsi sebagai shortcut bagi fungsi-fungsi yang sering diakses di Main Menu yang berwujud urutan tombol-tombol dengan ikon berupa gambar.
7. *Server Explorer*, berfungsi sebagai tempat manajemen database apabila aplikasi yang

dibuat di Visual Basic.NET terdapat transaksi databasenya. [6]

### 2.8 Teori Pengujian

Pengujian *software* adalah elemen kritis dari jaminan kualitas *software* dan merupakan *review* akhir dari spesifikasi, perancangan dan pengkodean. Pada tahap awal dari pengembangan *software*, *engineer* berusaha untuk membangun *software* dari sebuah konsep abstrak menjadi implementasi nyata. Pada saat pengujian, *engineer* membuat serangkaian kasus uji yang bertujuan untuk merusak *software* yang telah dibuat.

Beberapa aturan objektif dalam pengujian *software* antara lain:

- 1) Pengujian adalah proses eksekusi sebuah program untuk menemukan *error*.
- 2) Kasus uji yang baik adalah sesuatu yang bisa mengungkapkan kemungkinan yang tinggi untuk menemukan *error-error* yang tidak ditemukan sebelumnya.
- 3) Pengujian yang sukses adalah sesuatu yang bisa mengungkapkan *error* yang tidak ditemukan sebelumnya.

Pada saat melakukan pengujian terhadap suatu *software* ada beberapa prinsip pengujian yang harus diperhatikan, diantaranya:

- a. Semua pengujian harus terlacak ke kebutuhan *user*.
- b. Pengujian harus direncanakan jauh sebelum pengujian dimulai.
- c. Pengujian dimulai dari yang kecil dan mengarah kepada yang besar.
- d. Pengujian yang sempurna adalah tidak mungkin.
- e. Agar pengujian berjalan efektif, maka sebaiknya dilakukan oleh pihak ketiga yang netral.

Pengujian *white box* adalah sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedur.

*Basis Path Testing* merupakan metode untuk membuat perancang kasus uji bisa menurunkan sebuah ukuran kompleksitas logika dari sebuah perancangan prosedural, dan ukuran ini selanjutnya digunakan untuk menentukan sekelompok data dasar dari jalur eksekusi.

*Black Box Testing* memfokuskan pada kebutuhan fungsional dari *software*. Pengujian ini memperbolehkan *software engineer* menurunkan sejumlah input yang ditujukan untuk menguji kebutuhan fungsional dari suatu program.[7]

## III. ANALISIS DAN PERANCANGAN

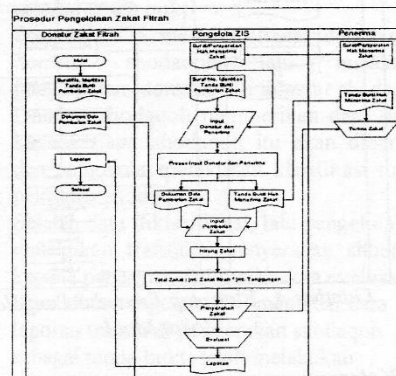
### 3.1 Analisis

Tahap analisis merupakan tahap pemahaman terhadap aplikasi yang dibuat. Pada tahap ini berisi hal-hal yang berhubungan dengan perangkat lunak yang digunakan, *Unified Modeling Language* (UML) dan perancangan *database*. Analisis bertujuan untuk mengetahui mekanisme sistem, proses-proses yang terlibat dalam sistem dan hubungan antara proses-proses tersebut.

#### 3.1.1 Analisis Sistem Berjalan

Untuk analisis sistem yang sedang berjalan, sistem yang dirancang adalah sebuah sistem aplikasi pengelolaan zakat, infaq dan shodaqoh pada suatu wilayah. Pada saat sekarang ini belum banyak aplikasi pengelolaan dana amal seperti amal zakat, infaq dan shodaqoh pada suatu masjid. Pengelolaannya masih menggunakan sistem *manual* dalam pengelolaan data-datanya.

##### 3.1.1.1 Flowmap Prosedur Pengelolaan Zakat Fitrah



Gambar 3 Flowmap Prosedur Pengelolaan Zakat Fitrah

#### Keterangan :

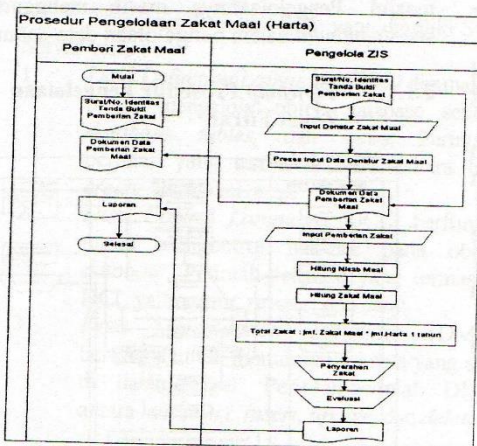
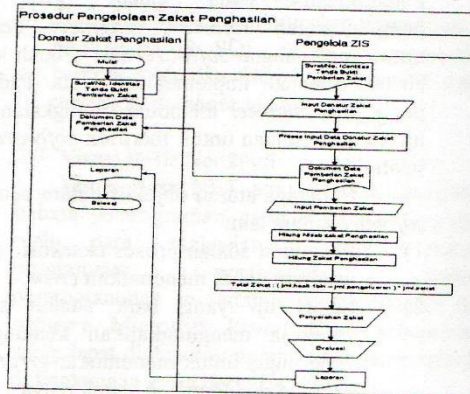
1. Donatur zakat fitrah mengajukan keterangan identitas diri pertanggungjawaban telah melakukan pemberian zakat fitrah sebagai tanda bukti.
2. Penerima zakat mengajukan keterangan identitas diri dan tanda bukti benar-benar layak mendapatkan dana yang telah di data sebelumnya.
3. Pengelola ZIS menginputkan data-data keterangan, baik donatur zakat fitrah maupun penerima zakat fitrah.

4. Donatur zakat fitrah akan menerima dokumen tanda bukti telah memberikan zakat fitrah.
5. Penerima zakat fitrah akan menerima dokumen tanda bukti berhak untuk menerima zakat.
6. Pengelola ZIS menghitung total zakat fitrah yang harus dibayar oleh donatur zakat.
7. Pengelola ZIS akan melakukan proses penyerahan zakat fitrah dan melakukan evaluasi.
8. Setelah melakukan penyerahan zakat fitrah dan melakukan evaluasi, lalu pengelola membuat laporan data donatur zakat fitrah, penerima zakat fitrah serta laporan transaksi penyerahan yang akan diserahkan kepada donatur zakat.

6. Setelah melakukan perhitungan total zakat maal dan melakukan evaluasi, lalu pengelola membuat laporan data donatur zakat maal yang akan diserahkan kepada donatur zakat.

### 3.1.1.3 Flowmap Prosedur Pengelolaan Zakat Penghasilan

### 3.1.1.2 Flowmap Prosedur Pengelolaan Zakat Maal



Gambar 5 Flowmap Prosedur Pengelolaan Zakat Penghasilan

Gambar 4. Flowmap Prosedur Pengelolaan Zakat Maal

#### Keterangan :

#### Keterangan :

1. Donatur zakat maal mengajukan keterangan identitas diri pertanggungjawaban telah melakukan pemberian zakat maal sebagai tanda bukti.
2. Pengelola ZIS menginputkan data-data keterangan donatur zakat maal.
3. Donatur zakat maal akan menerima dokumen tanda bukti telah memberikan zakat maal.
4. Penentuan berapa nisab dari zakat maal yang akan dibayar oleh donatur zakat.
5. Pengelola ZIS menghitung total zakat maal yang harus dibayar oleh donatur zakat dengan rumus perhitungannya adalah :

1. Donatur zakat penghasilan mengajukan keterangan identitas diri pertanggungjawaban telah melakukan pemberian zakat penghasilan sebagai tanda bukti.
2. Pengelola ZIS menginputkan data-data keterangan donatur zakat penghasilan.
3. Donatur zakat penghasilan akan menerima dokumen tanda bukti telah memberikan zakat penghasilan.
4. Penghitungan berapa nisab dari zakat penghasilan yang akan dibayar oleh donatur zakat.
5. Pengelola ZIS menghitung total zakat penghasilan yang harus dibayar oleh donatur zakat dengan rumus perhitungannya adalah :

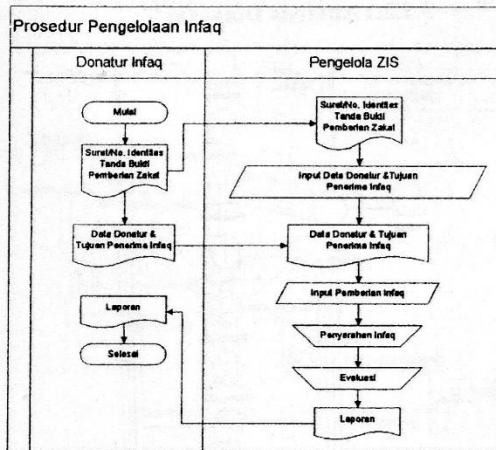
$$\frac{(\text{Jml.Hasil 1 bln} - \text{Jml.pengeluaran}) *}{\text{Jml.Zakat}}$$

6. Setelah melakukan perhitungan total zakat penghasilan dan melakukan evaluasi, lalu pengelola membuat laporan data donatur zakat penghasilan yang akan diserahkan kepada donatur zakat.

$$\text{Jml.Harta 1 tahun} * \text{jml.zakat maal}$$



### 3.1.1.4 Flowmap Prosedur Pengelolaan Infaq

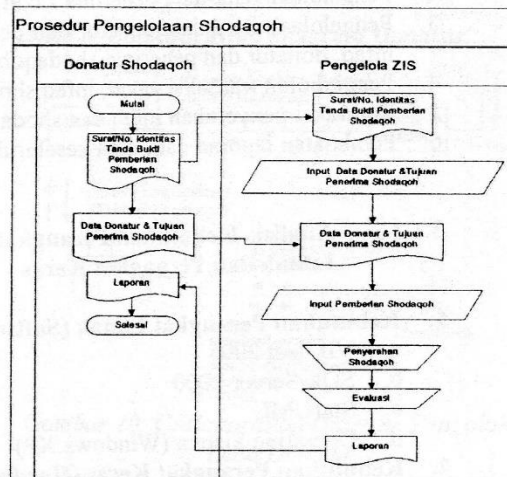


Gambar 6 Flowmap Prosedur Pengelolaan Infaq

**Keterangan :**

1. Donatur infaq mengajukan identitas keterangan diri pertanggungjawaban telah melakukan transaksi pemberian infaq sebagai tanda bukti.
2. Pengelola ZIS mengklarifikasi identitas diri donatur infaq lalu melakukan penginputan identitas diri donatur infaq.
3. Donatur infaq memberikan data tujuan kepada siapa infaq itu akan diberikan dan pengelola melakukan klarifikasi tujuan penerima infaq.
4. Setelah data diklarifikasi, lalu pengelola ZIS melakukan transaksi penyerahan infaq kepada penerima dan melakukan evaluasi.
5. Kemudian pengelola ZIS membuat data laporan transaksi penyerahan infaq sebagai tanda bukti telah melakukan transaksi dan diberikan kepada donatur infaq.

### 3.1.1.5 Flowmap Prosedur Pengelolaan Shodaqoh



Gambar 7 Flowmap Prosedur Pengelolaan Shodaqoh

Gambar 3.5 Flowmap Prosedur Pengelolaan Shodaqoh

**Keterangan :**

1. Donatur shodaqoh mengajukan identitas keterangan diri pertanggungjawaban telah melakukan transaksi pemberian shodaqoh sebagai tanda bukti.
2. Pengelola ZIS mengklarifikasi identitas diri donatur shodaqoh lalu melakukan penginputan identitas diri donatur shodaqoh.
3. Donatur shodaqoh memberikan data tujuan kepada siapa shodaqoh itu akan diberikan dan pengelola melakukan klarifikasi tujuan penerima shodaqoh.
4. Setelah data diklarifikasi, lalu pengelola ZIS melakukan transaksi penyerahan shodaqoh kepada penerima dan melakukan evaluasi.
5. Kemudian pengelola ZIS membuat data laporan transaksi penyerahan shodaqoh sebagai tanda bukti telah melakukan transaksi dan diberikan kepada donatur shodaqoh.

### 3.1.2 Analisis Sistem yang akan Dibangun

#### 3.1.2.1 Analisis Kebutuhan Aplikasi

Apabila ketika ada suatu transaksi penginputan zakat, baik zakat fitrah, zakat maal (harta) atau pun zakat penghasilan, penginputan infaq dan juga penginputan shodaqoh, pengelola ZIS akan mulai menjalankan kegiatan pengelolaan dari setiap transaksi. Pengelola ZIS tersebut menggunakan aplikasi pengelolaan Zakat, Infaq dan Shodaqoh dalam mengelola setiap transaksi. Dalam aplikasi ini, proses-proses yang dapat dilakukan antara lain, yaitu :

1. Pengelolaan data-data donatur zakat fitrah, zakat maal dan zakat penghasilan.

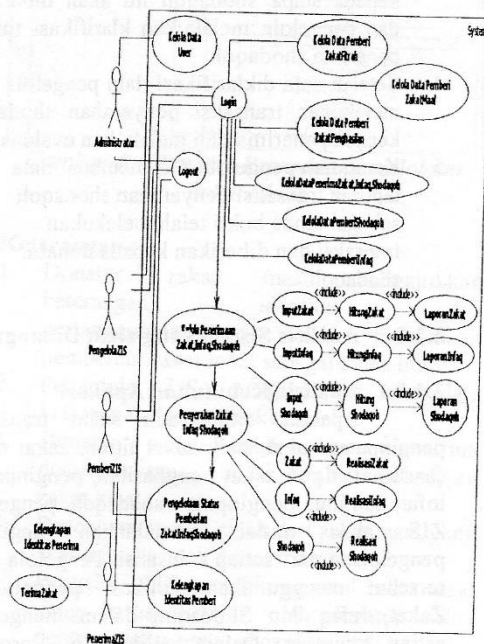
2. Pengelolaan data-data penerima zakat.
3. Pengelolaan data-data donatur dan penerima infaq, donatur dan penerima shodaqoh
4. Penginputan transaksi zakat, infaq shodaqoh
5. Transaksi penyerahan infaq dan shodaqoh
6. Pembuatan laporan data-data keseluruhan

### 3.1.2.2 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

1. **Kebutuhan Perangkat Lunak (Software)**
  - a. VB. Net 2005
  - b. SQL Server 2000
  - c. StarUML
  - d. Operating Sistem (Windows XP)
2. **Kebutuhan Perangkat Keras (Hardware)**
  - a. Prosesor : Intel Pentium Dual-Core
  - b. RAM: 512 MB
  - c. VGA: 256 MB
  - d. 120 GB HDD
  - e. Hardware pendukung lainnya: Mouse, dan Speaker, Switch, Kabel RJ45.

### 3.2.1 Usecase Diagram

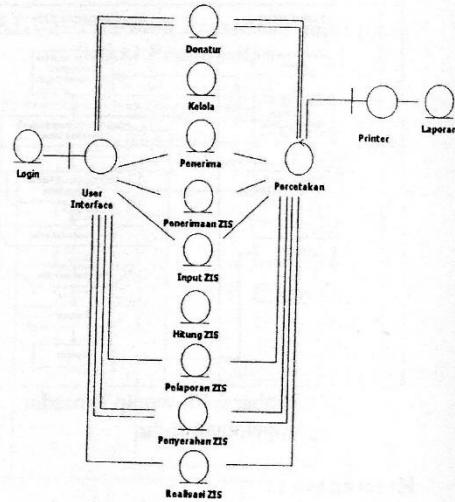
Use case diagram menggambarkan interaksi antara use case dan aktor dalam suatu sistem. Untuk mempermudah pemodelan, use case diagram untuk Sistem Aplikasi Website Objek Wisata Kabupaten Lingga adalah seperti dibawah ini :



Gambar 7 UseCase Diagram

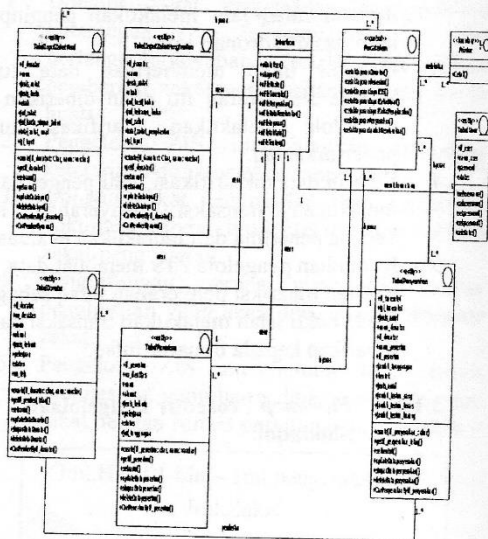
### 3.2.2 Design Diagram

#### 3.2.2.1 Analisis Diagram



Gambar 3. 7 Analisis Diagram

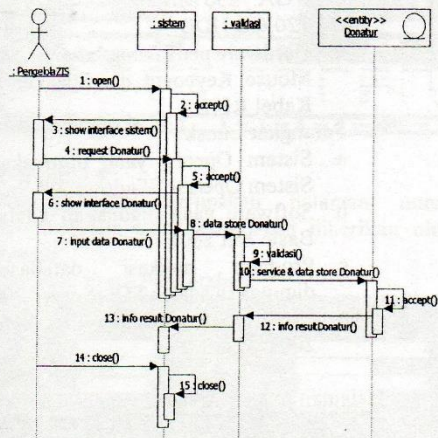
#### 3.2.2.2 Class Diagram



Gambar 8 Class Diagram.

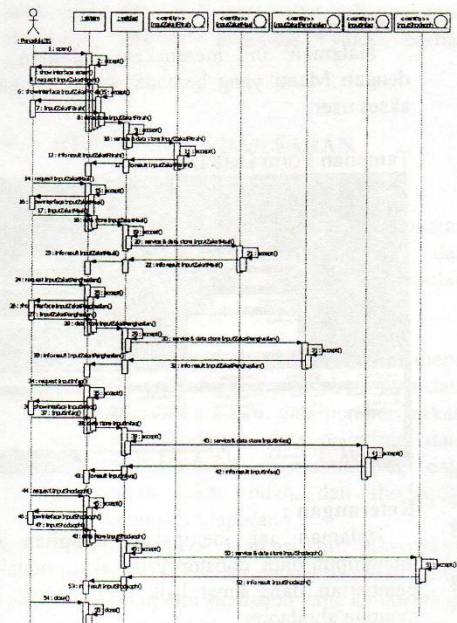
### 3.2.3 Interaction Diagram

#### 3.2.3.1 Sequence Diagram Pengolahan Data Donatur



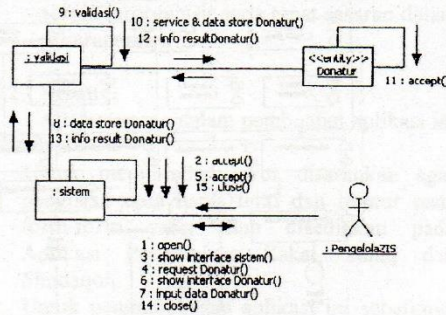
Gambar 8 Sequence Donatur

#### 3.2.3.2 Sequence Diagram Input Zakat Fitrah, Zakat Maal, Zakat Penghasilan, Input Infaq dan Input Shodaqoh



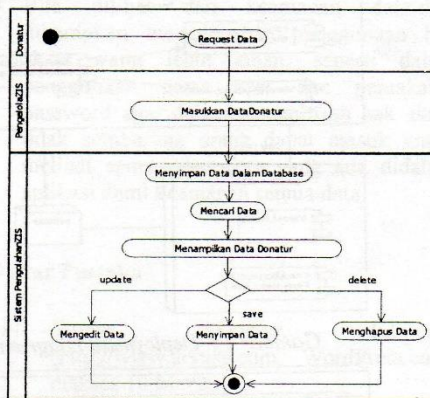
Gambar 9 Sequence Diagram Input Zakat Fitrah, Zakat Maal, Zakat Penghasilan, Input Infaq dan Input Shodaqoh

#### 3.2.3.3 Collaboration Diagram Donatur



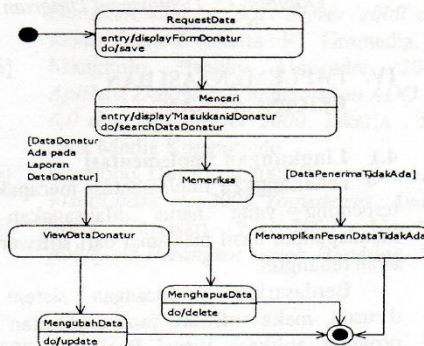
Gambar 10 Collaboration Diagram Pengolahan Data Donatur

### 3.2.4 Activity Diagram Donatur



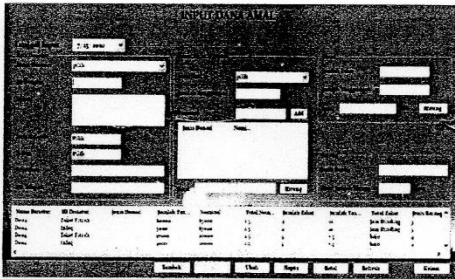
Gambar 11 Activity Diagram Pengelolaan Data Donatur

### 3.2.5 Statechart Diagram Donatur



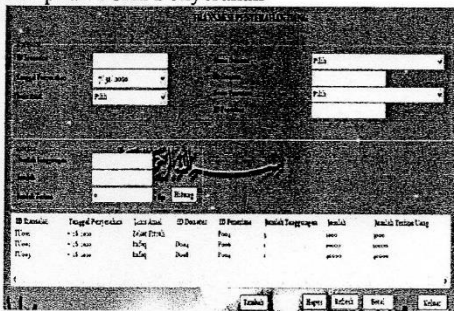
Gambar 12 Statechart Diagram Pengelolaan Data Donatur





**Keterangan :**  
Halaman ini merupakan halaman untuk menginputkan dana yang akan diberikan oleh donatur.

**Tampilan Form Penyerahan**



**Keterangan :**  
Halaman ini merupakan halaman melakukan penyerahan dana baik dana zakat, infaq maupun shodaqoh kepada penerima yang berhak menerimanya.

**V. KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

Berdasarkan hasil evaluasi pada pembuatan Aplikasi Pengelolaan Zakat, Infaq dan Shodaqoh, maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi ini menyediakan menu-menu dalam bentuk form pengolahan data-data suatu input antara lain: data pemberi zakat, infaq dan shodaqoh, data penerima, data input zakat, infaq dan shodaqoh, data penyerahan zakat, infaq, dan shodaqoh, serta pembuatan laporan.
2. Dengan adanya aplikasi ini diharapkan mempermudah suatu pengelolaan data-data zakat, infaq dan shodaqoh yang kebanyakan masih memakai cara manual dalam penyimpanan datanya
3. Diharapkan pula aplikasi ini dapat lebih mengefesienkan lagi dalam melakukan

pendataan pemberi zakat, infaq dan shodaqoh maupun penerima yang tersusun secara terorganisir serta tepat sasaran dalam penerapannya.

**5.2 Saran**

Adapun saran dalam pembuatan aplikasi ini adalah :

1. Untuk menghindari error disarankan agar pengisian data harus teliti dan teratur pada form-form yang telah disediakan pada Aplikasi Pengelolaan Zakat, Infaq dan Shodaqoh.
2. Untuk pengembangan aplikasi ini sebaiknya tidak hanya digunakan untuk pengelolaan data transaksi zakat, infaq maupun shodaqoh saja, tetapi juga diharapkan untuk transaksi amal lainnya baik dalam bentuk keuangan maupun dalam bentuk benda, agar semua yang didata benar-benar tersusun rapih.
3. Bila dilihat dari keamanan data-data diharapkan menggunakan penggunaan hak akses yang lebih aman, seperti dalam penggunaan nama user dan pemakaian password agar dalam penggunaan hak akses tidak sembarang orang dapat masuk untuk melihat semua data-data yang ada didalam aplikasi demi keamanan semua data.

**Daftar Pustaka**

- [1] *Sistem Informasi*, (Online), (<http://www.google.com>, WordPress.com, diakses 10 Mei 2010).
- [2] Darmayuda, Ketut. 2007. *Program Aplikasi Client Server*. Bandung : Informatika Bandung.
- [3] Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- [4] Budiharto, Widodo. 2004. *Aplikasi Database dengan SQL Server 2000 dan Visual Basic 6*. Jakarta : PT. Gramedia.
- [5] Mangkulo, Hengky Alexander. 2004. *Aplikasi Database Menggunakan ADO VB 6.0 dan SQL Server 2000*. Jakarta : PT. Elex Media Komputindo
- [6] Hardjono, Dewiberta.2006. *Pemrograman Visual Basic.Net 2005*. Yogyakarta : Andi.
- [7] Sofia Fitriyani, Novi.2009. *Handout Rekayasa Perangkat Lunak*. Bandung.