



TUGAS AKHIR

DESAIN SISTEM INFORMASI PENYALUR DANA ALOKASI UMUM NAGARI (DAUN) PADA BAGIAN PEMERINTAH NAGARI DAN URUSAN RANTAU DI PEMERINTAHAN DAERAH KABUPATEN TANAH DATAR

*Diajukan Kepada Program D.III Manajemen Informatika
Untuk Memenuhi Salah Satu Syarat Guna Mencapai Gelar Ahli Madya
Dalam Bidang Ilmu Manajemen Informatika*

Oleh :
ROBBY WAHYUDI
NIM. 10 205 059

**PROGRAM D.III MANAJEMEN INFORMATIKA
SEKOLAH TINGGI AGAMA ISLAM NEGERI (STAIN)
BATUSANGKAR
2014**

PENGESAHAN TIM PENGUJI

Tugas Akhir yang berjudul “**DESAIN SISTEM INORMASI PENYALUR DANA ALOKASI UMUM NAGARI PADA PEMERINTAH NAGARI DAN URUSAN RANTAU DI PEMERINTAHAN DAERAH KABUPATEN TANAH DATAR**” oleh **ROBBY WAHYUDI NIM 10 205 059**, telah diujikan pada Sidang Komprehensif Sekolah Tinggi Agama Islam Negeri (STAIN) Batusangkar, Jum’at 07 Februari 2014 dan dinyatakan telah dapat diterima sebagai salah satu syarat untuk mencapai gelar Ahli Madya Program Diploma III (D.III) Manajemen Informatika.

Batusangkar, 10 Februari 2014

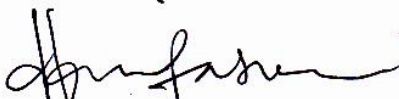
Tim Penguji Sidang Komprehensif
Ketua/Sekretaris



ISWANDI, M.Kom

NIP. 1970 0510 2003 12 1 004

Anggota



LITA SARI MUCHLIS, M.kom

NIP.19780122200801 2 017



ALFONSO FN, S.KOM

Mengetahui

Ketua Program Studi D.III Manajemen Informatika
STAIN Batusangkar



ISWANDI, M.Kom

NIP. 1970 0510 2003 12 1 004

KATA PENGANTAR



Puji syukur penulis haturkan atas kehadiran Allah SWT. yang telah melimpahkan rahmat dan karunia-Nya sehingga pembuatan tugas akhir ini dapat terselesaikan. Salawat dan salam buat junjungan umat, yakni Nabi Muhammad SAW. yang telah membawa manusia dari alam kebodohan ke alam yang penuh dengan ilmu pengetahuan seperti kita rasakan seperti sekarang ini.

Maksud dan tujuan pembuatan tugas akhir ini adalah untuk memberikan sumbangan pemikiran kepada almamater serta untuk memenuhi sebagian persyaratan untuk mencapai derajat Diploma III Program Studi Manajemen Informatika Sekolah Tinggi Agama Islam Negeri Batusangkar.

Dalam penulisan tugas akhir ini penulis banyak mendapat bimbingan dan bantuan baik moril maupun materil dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Prof. Dr. H. Hasan Zaini, MA selaku ketua STAIN Batusangkar.
2. Bapak Drs. Syamsuwir, M.Ag selaku Ketua Jurusan Syariah STAIN Batusangkar.
3. Bapak Iswandi, M.Kom selaku ketua Program Studi Manajemen Informatika STAIN Batusangkar dan Pembimbing yang telah meluangkan waktu dan tenaga dalam membimbing penulis dalam menyelesaikan penulisan Tugas Akhir ini.
4. Ibuk Roza Melfita, S.STP selaku Bendahara Umum serta seluruh pegawai yang telah membantu penulis selama mengumpulkan data yang diperlukan dalam penyusunan Tugas Akhir ini.
5. Ayah dan Ibu, serta Keluarga maupun sanak famili penulis, yang telah memberikan bantuan baik moril maupun materil.
6. Serta seluruh rekan-rekan MI angkatan 10 yang senasib dan seperjuangan dengan penulis.

7. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan motivasi dan semangat serta sumbangan pemikirannya kepada penulis sehingga selesainya laporan ini.

Penulis sadar bahwasanya tugas akhir ini jauh dari kesempurnaan, oleh karena itu kritikan dan saran yang sifatnya membangun sangat penulis harapkan untuk kesempurnaan tugas akhir ini. Penulis juga berharap semoga penulisan tugas akhir ini memberikan manfaat kepada kita semua. Amin...

Akhirnya kepada Allah SWT jualah penulis bermohon dan bersujud semoga keikhlasan yang diberikan akan dibalas-Nya. *Amin Ya Robbal'alamin*.

Batusangkar, Januari 2014

Penulis

ABSTRAK

JUDUL TUGAS AKHIR : DESAIN SISTEM INFORMASI PENYALUR DANA ALOKASI UMUM NAGARI (DAUN) PADA BAGIAN PEMERINTAH NAGARI DAN URUSAN RANTAU DI PEMERINTAHAN DAERAH KABUPATEN TANAH DATAR

NAMA MAHASISWA : ROBBY WAHYUDI

NOMOR INDUK : 10 205 059

PROGRAM STUDI : MANAJEMEN INFORMATIKA

DOSEN PEMBIMBING : ISWANDI, M.Kom

Berdasarkan penelitian yang telah dilakukan pada Pemerintahan Nagari dan Urusan Rantau, maka ditemukan permasalahan mengenai pengolahan laporan keuangan, yang mana pengolahannya masih menggunakan *Microsoft Excel* dan *Microsoft Word*, yang ditinjau dari segi keefektif dan keefesiennya mengakibatkan setiap data yang diperlukan belum akurat, cepat dan tepat waktu.

Dalam penulisan tugas akhir ini metode penelitian yang dilakukan adalah penelitian lapangan yaitu wawancara dengan mengajukan pertanyaan dengan melalui tanya jawab, penelitian perpustakaan dan penelitian di labor dengan menggunakan pemrograman Visual Basic 10.

Dengan memanfaatkan sistem komputerisasi diharapkan dapat membantu proses pengambilan keputusan dan dengan memanfaatkan Visual Basic 10 sebagai software aplikasi diharapkan dapat menggantikan cara yang kurang efisien dan efektif serta diharapkan dapat mempermudah dalam pengolahan laporan keuangan DAUN (Dana Alokasi Umum Nagari) pada Pemerintah Nagari dan Urusan Rantau.

Kata Kunci : Pengolahan, Laporan Keuangan DAUN, Pemerintahan Nagari dan Urusan Rantau, Akurat

DAFTAR ISI

HALAMAN PENGESAHAN

	Hal
KATA PENGANTAR	i
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix

BAB I PENDAHULUAN

A. Latar Belakang Masalah	1
B. Identifikasi Masalah	4
C. Batasan Masalah	4
D. Rumusan Masalah.....	5
E. Tujuan Penelitian	5
F. Kegunaan Penelitian	5
G. Metode Penelitian	6
H. Sistematika Penulisan	6

BAB II LANDASAN TEORI

A. Gambaran Umum	8
1. Sejarah	8
2. Tugas Pokok dan Fungsi.....	9
3. Tugas dan Tanggung Jawab.....	11
4. Struktur Organisasi	15
5. Visi dan Misi.....	15
B. Konsep Dasar Sistem Informasi	16
1. Sistem	16
a. Pengertian Sistem.....	16

b. Syarat-syarat Sistem	17
c. Pengelompokan Sistem	18
2. Informasi	18
a. Pengertian Informasi	18
b. Pengguna Informasi	19
3. Sistem Informasi	20
a. Pengertian Sistem Informasi	20
b. Konponen Sistem Informasi	21
4. Perancangan Sistem Informasi	21
a. Pengertian Perancangan Sistem	21
b. Tujuan Perancangan Sistem	22
C. Alat Bantu Perancangan Sistem	22
1. Aliran Sistem Informasi	22
2. Data Flow Diagram (DFD)	23
3. Entity Relationship Diagram (ERD)	24
4. Program Flowchart	25
5. Normalisasi	27
D. Mengenal Database	28
1. Pengertian Database	28
E. Mengenal Microsoft Access 2010	29
1. Memulai Microsoft Access 2010	29
2. Komponen Microsoft Access 2010	30
3. Mengenal Database Microsoft Access 2010	31
4. Tipe Data	31
F. Mengenal Microsoft Visual Basic 2010	34
1. Sejarah	34
2. Memulai Visual Basic 2010	35
3. Konponen Visual Basic 2010	36
a. Jendela Utama	37

b. Jendela Form.....	38
c. Project Windows	39
d. Toobox.....	39
e. Jendela Properti.....	40
f. Jendela Kode	41
g. Event.....	42
h. Method.....	42
i. Module.....	42
4. Struktur Program Visual Studio 2010.....	42
a. Bagian Deklarasi	43
a. Bagian Pernyataan.....	43
5. Tipe Data Pemograman Visual Basic 2010	44

BAB III ANALISA DAN HASIL

A. Analisa Sistem yang Sedang Berjalan	45
1. Bagan Alir Sistem (ASI)	45
2. Evaluasi Sistem yang Sedang Berjalan	47
B. Desain Sistem Baru	48
1. Desain Global	48
a. Aliran Sistem Informasi Baru	48
b. Contex Diagram.....	50
c. Data Flow Diagram (DFD)	50
d. Entity Relationship Diagram (ERD).....	52
e. Struktur Program.....	52
2. Desain Terinci	53

	a. Desain Output	53
	b. Desain Input	56
	c. Desain File	58
BAB IV	PENUTUP	
	A. Kesimpulan	60
	B. Saran	61
DAFTAR PUSTAKA		
LAMPIRAN		
	- FLOWCHART	
	- LISTING PROGRAM	

DAFTAR GAMBAR

Gambar 2.1	Struktur Organisasi.....	15
Gambar 2.2	Siklus Informasi	19
Gambar 2.3	Konsep Sistem Informasi	20
Gambar 2.4	Tampilan Awal Microsoft Access.....	30
Gambar 2.5	Halaman StarUp	36
Gambar 2.6	Tampilan Pertama Microsoft Visual Basic 2010	37
Gambar 2.7	Main Windows	38
Gambar 2.8	Form Windows	38
Gambar 2.9	Solution Explorer	39
Gambar 2.10	Toolbox	40
Gambar 2.11	Jendela Properties.....	41
Gambar 2.12	Jendela Kode	42
Gambar 3.1	Bagan Alir Dokumen Yang Sedang Berjalan	47
Gambar 3.2	Bagan Alir Dokumen Yang diusulkan	49
Gambar 3.3	Context Diagram	50
Gambar 3.4	Data Flow Diagram (DFD)	51
Gambar 3.5	Entity Relationship Diagram (ERD)	52
Gambar 3.6	Struktur Program	53
Gambar 3.7	Desain Input Kecamatan	56
Gambar 3.8	Desain Input Nagari	57
Gambar 3.9	Desain Input Transaksi DAUN	58

DAFTAR TABEL

Tabel 2.1	Simbol Aliran Sistem Informasi	23
Tabel 2.2	Simbol Data Flow Diagram (DFD).....	24
Tabel 2.3	Simbol Entity Relationship Diagram (ERD)	25
Tabel 2.4	Program Flowchart.....	26
Tabel 2.5	Tipe Data Microsoft Access.....	31
Table 2.6	Tipe Number	33
Tabel 2.7	Tipe Data Microsoft Visual Basic 2010.....	44
Tabel 3.1	Tabel Laporan Data Kecamatan.....	54
Tabel 3.2	Tabel Laporan Data Nagari.....	54
Tabel 3.3	Tabel Laporan Data Transaksi DAUN.....	55
Tabel 3.4	Tabel Transaksi DAUN	58
Tabel 3.5	Tabel Transaksi Kecamatan	59
Tabel 3.6	Tabel Transaksi Nagari	69

BAB 1

PENDAHULUAN

A. Latar Belakang Masalah

Di era yang semakin maju, teknologi informasi pun semakin berkembang dengan cepat. Bahkan hampir setiap pekerjaan memanfaatkan komputer untuk mempermudah pekerjaan tersebut. Sebagai contoh adalah laporan keuangan pada pemerintahan nagari. Seiring bertambah besarnya jumlah keuangan nagari maka perlu memberikan pelayanan berupa sistem informasi.

Pemerintahan Nagari merupakan salah satu institusi pelayanan masyarakat dalam melaksanakan kegiatan yang berhubungan dengan masyarakat nagari. Pemberian layanan dalam banyak hal akan mempengaruhi kegiatan di Pemerintahan nagari. Kegiatan tersebut tidak akan berjalan dengan baik jika masih menggunakan sistem yang manual. Maka dibutuhkan pengolahan data secara komputerisasi dan terprogram.

Sistem informasi pada Pemerintahan Nagari merupakan suatu pengelolaan informasi di seluruh seluruh tingkat pemerintah secara sistematis dalam rangka penyelenggaraan pemerintahan nagari yang aksesnya lebih cepat, akurat dan tepat kepada masyarakat.

Akses berarti ruang dan kapasitas masyarakat untuk masuk dalam arena *governance*, yakni mempengaruhi dan menentukan kebijakan serta terlibat aktif mengelola barang-barang publik. Akses warga terhadap pelayanan publik termasuk dalam pembahasan ini.

Ada dua hal penting dalam akses: keterlibatan secara terbuka (*inclusion*) dan keikutsertaan (*involvement*). Keduanya mengandung kesamaan tetapi berbeda titik tekannya. *Inclusion* menyangkut siapa yang terlibat, sedangkan *involvement* berbicara tentang bagaimana masyarakat terlibat. Keterlibatan berarti ketersediaan ruang dan kemampuan bagi siapa

saja untuk terlibat dalam proses politik, terutama kaum miskin, minoritas, rakyat kecil, perempuan, dan lain-lain.

Akses akan menjadi arena titik temu antara warga dan pemerintah. Pemerintah wajib membuka ruang akses warga dan memberikan layanan publik pada warga, terutama kelompok-kelompok marginal. Sebaliknya warga secara bersama-sama proaktif mengidentifikasi problem, kebutuhan dan potensinya maupun merumuskan gagasan, pemecahan masalah dan pengembangan potensi secara sistematis. Pemerintah wajib merespons gagasan warga sehingga bisa dirumuskan visi dan kebijakan bersama dengan berpijak pada kemitraan dan kepercayaan.

Self-control ini sangat penting karena masyarakat sudah lama berada dalam konteks penindasan berantai: yang atas menindas yang bawah, sementara yang paling bawah saling menindas ke samping. Artinya kontrol eksternal digunakan masyarakat untuk melawan eksploitasi dari atas, sementara self-control dimaksudkan untuk menghindari mata rantai penindasan sesama masyarakat, seraya hendak membangun tanggungjawab sosial, komitmen dan kompetensi warga terhadap segala sesuatu yang mempengaruhi kehidupannya sehari-hari. Supaya partisipasi masyarakat dalam pembangunan ini tetap eksis, maka pemberdayaan masyarakat harus terus dikembangkan.

Dalam hal partisipasi masyarakat dalam perencanaan pembangunan, Pemerintah Daerah telah berupaya mengikutkan sertakan seluruh lapisan masyarakat ikut berpartisipasi (sato sakaki) dalam perencanaan pembangunan. Hal ini dapat diwujudkan dalam pelaksanaan Musrenbang yang dimulai dari tingkat Nagari, Kecamatan dan Kabupaten dengan melibatkan anggota DPRD pada Musrenbang tersebut.

Dalam upaya merangsang partisipasi masyarakat dalam pembangunan serta peningkatan ekonomi masyarakat, dilakukan beberapa upaya diantaranya pengalokasian dana partisipasi nagari yang disesuaikan

dengan luas wilayah dan potensi nagari yang bersangkutan, pelaksanaan dana tabungan nagari, pengalokasian dana stimulan, pemberian modal pinjaman langsung kepada masyarakat dan lain-lain. Dengan adanya Dana Penunjang Partisipasi Pembangunan tersebut diharapkan partisipasi masyarakat makin meningkat dari tahun ke tahun, dalam rangka menggali dan mengembangkan seluruh potensi yang ada pada setiap nagari sehingga akan mampu meningkatkan partisipatif masyarakat dalam pembangunan.

Dalam rangka peningkatan pemerataan pembangunan pada seluruh aspek guna mewujudkan kesejahteraan dan kemakmuran masyarakat Nagari dalam Kabupaten Tanah Datar, Pemerintah Daerah Kabupaten Tanah Datar memberikan Bantuan Dana Alokasi Umum Nagari (DAUN) bagi 75 (Tujuh puluh lima) Nagari pada 14 (Empat belas). Dari Dana Alokasi Umum Nagari (DAUN) yang ada, Pemerintah menyadari sepenuhnya tidak semua kegiatan pembangunan Nagari dapat dibiayai dengan dana tersebut. Sebagai salah satu solusi dari permasalahan diatas, Pemerintah Daerah Kabupaten Tanah Datar mengeluarkan kebijakan memberikan bantuan Dana Penunjang Partisipatif Nagari dengan tujuan memberikan kesempatan/ peluang kepada wadah/ unsur yang ada di Nagari untuk mengembangkan semangat swadaya partisipasi pembangunan.

Demi tertibnya pelaksanaan Dana Penunjang Partisipatif Nagari diatas, secara administratif, maka Pemerintah Daerah Kabupaten Tanah Datar mengeluarkan Peraturan Bupati No 16 tentang Pedoman Penyusunan Anggaran Pendapatan Belanja Nagari sebagai pedoman / acuan bagi Nagari dan lembaga terkait pengguna dana dalam pelaksanaan tertib administrasi, pengajuan dan penggunaan dana yang ada pada Nagari.

Pengolahan dana yang ada pada nagari di Kabupaten Tanah Datar menggunakan sistem manual, sehingga memerlukan waktu yang lebih lama. Tentu hal ini akan sangat menyulitkan apabila dana, data atau informasi tentang Pemerintahan nagari yang dibutuhkan semakin banyak. Untuk

mengatasi masalah tersebut, maka Penulis mencoba merancang sebuah sistem informasi untuk membantu Pemerintahan Nagari dan Urusan Rantau khususnya pengolahan dana dalam nagari dalam sebuah Tugas Akhir dengan judul “**DESAIN SISTEM INFORMASI PENYALUR DANA ALOKASI UMUM NAGARI (DAUN) PADA BAGIAN PEMERINTAH NAGARI DAN URUSAN RANTAU DI PEMERINTAHAN DAERAH KABUPATEN TANAH DATAR**”

B. Identifikasi Masalah

Berdasarkan latar belakang permasalahan yang ada pada Bagian Pemerintahan Nagari dan Urusan Rantau Kabupaten Tanah Datar di atas maka Penulis dapat mengidentifikasi masalah yang dapat dijadikan sebagai acuan dalam melaksanakan pengolahan data sebagai berikut :

1. Pengolahan Dana Alokasi Umum Nagari (DAUN) masih dilakukan dengan manual.
2. Belum adanya sistem informasi yang didukung oleh bahasa pemrograman yang dilengkapi dengan *database*, yang mengakibatkan terjadinya penumpukan arsip yang akan merepotkan pegawai.

C. Batasan Masalah

Berdasarkan identifikasi masalah di atas serta mengingat keterbatasan waktu yang tersedia, maka Penulis membatasi permasalahan tentang penyalur Dana Alokasi Umum Nagari (DAUN) pada Pemerintah Nagari dan Urusan rantau di Pemerintahan Kabupaten Tanah Datar.

D. Rumusan Masalah

Berdasarkan batasan masalah di atas, maka Penulis dapat merumuskan masalah yaitu: Bagaimana Sistem informasi penyalur Dana Alokasi Umum Nagari (DAUN) pada Pemerintah Nagari dan Urusan Rantau.

E. Tujuan Penelitian

Sistem informasi penyalur keuangan DAUN ini dirancang dengan tujuan:

1. Untuk mengetahui dan mengenali sistem yang ada pada Bagian Pemerintahan Nagari dan Urusan Rantau.
2. Untuk membantu pengolahan data keuangan DAUN dan untuk merancang sebuah sistem informasi yang dapat membantu Bagian Pemerintahan Nagari dan Urusan Rantau.
3. Mempermudah dalam memantau penyalur Dana Alokasi Umum Nagari (DAUN) pada Bagian Pemerintahan Nagari dan Urusan Rantau.

F. Kegunaan Penelitian

1. Sebagai implementasi dan pengembangan ilmu yang telah Penulis dapatkan selama masa perkuliahan.
2. Sebagai bahan kajian dan masukan bagi pegawai Pemerintahan Nagari dan Urusan Rantau.
3. Sebagai tambahan referensi bagi pembaca yang akan melakukan penelitian lebih lanjut.
4. Sebagai persyaratan untuk memperoleh gelar Ahli Madya Program Diploma III (D.3) Manajemen Informatika pada STAIN Batusangkar.

G. Metode Penelitian

Dalam penulisan ini penulis menggunakan beberapa metode, antara lain :

1. Penelitian Lapangan (*Field Research*)

Dalam penelitian ini Penulis mendapat data langsung dari hasil peninjauan ke lapangan, yaitu Pemerintah Kabupaten Tanah Datar, melalui wawancara pada bagian-bagian yang berhubungan langsung terhadap pembahasan yang diteliti.

2. Penelitian Kepustakaan (*Library Research*)

Penelitian ini dilakukan untuk mencari, mengumpulkan data, mempelajari data dari buku-buku, bahan kuliah, karangan ilmiah, ataupun tulisan yang berhubungan dengan permasalahan yang dibahas dalam penelitian ini.

3. Penelitian Laboratorium (*Laboratory Research*)

Penulis melakukan pengolahan data yang telah dikumpulkan dengan menggunakan *Laptop* yang telah dilengkapi *hardware* dan *software* sebagai alat bantu dalam pembuatan tugas akhir ini.

H. Sistematika Penulisan

Penulisan ini dilakukan dengan membagi penjelasannya pada beberapa BAB, dimana tahap dan struktur dari penulisannya dapat dilihat sebagai berikut:

BAB I yang merupakan Pendahuluan yang terdiri dari : Latar Belakang, Identifikasi Masalah, Batasan Masalah, Rumusan Masalah, Tujuan, Kegunaan, Metodologi Penelitian dan Sistematika Penulisan.

BAB II yang merupakan Landasan Teori terdiri dari, Gambaran Umum Bagian Pemerintahan Nagari dan Urusan Rantau, Sekilas Tentang Data, Sekilas Tentang Kesehatan, Konsep Dasar Sistem Informasi, dan Pengenalan *Microsoft Visual Studio Ultimate*.

BAB III merupakan Analisa dan Hasil, terdiri dari Analisa Sistem yang Sedang Berjalan dan Perancangan Sistem Baru yang terdiri dari Perancangan Global dan Perancangan Terinci serta Normalisasi.

BAB IV terdiri dari Kesimpulan dan Saran. Bab ini berisi kesimpulan seluruh rangkaian kegiatan selama proses penelitian yang menyatakan kelebihan dan kekurangan dari hasil selama penelitian berlangsung, serta saran-saran bagi perbaikan dan pengembangan sistem yang dimungkinkan.

BAB II

LANDASAN TEORI

A. Gambaran Umum

1. Sejarah dan Keadaan Geografis

Kabupaten Tanah Datar adalah salah satu kabupaten di Propinsi Sumatera Barat yang dikenal sebagai “Luhak Nan Tuo”. Wilayah administrasi Kabupaten Tanah Datar terdiri dari 14 Kecamatan dan pada level Nagari (setingkat Kelurahan) terdapat 75 nagari.

Pada tahun 1988 Kantor Bupati Tanah Datar mendapat jatah gedung baru yang terletak di Pagaruyung Kecamatan Tanjung Emas. Dengan Kepala Daerah pertama dipimpin jabatan oleh Sidi Bakarudin,SH dilantik pada tanggal 01 Maret 1945 sampai dengan 21 Januari 1951. Bupati kedua Tanah Datar di pimpin oleh Ibrahim DT Pamuncak masa periode 1951 sampai dengan 1959. Bupati Tanah Datar ketiga di pimpin oleh Soetoro dengan masa jabatan 1959 sampai dengan 1961. Bupati Tanah Datar keempat dipimpin oleh Mahyudin Al Gamar masa periode 1961 sampai dengan 1874. Bupati Tanah Datar kelima di pimpin oleh Sulaiman dengan masa periode 1974 sampai dengan tahun 1980. Bupati Tanah Datar keenam di pimpin oleh Muhamad Nalis dengan masa periode Tahun 1980 sampai dengan Tahun 1985. Bupati Tanah Datar ketujuh di kepalai oleh Ikasuma Hamid dengan masa periode dari Tahun 1985 sampai dengan Tahun 1995. Bupati Tanah Datar kedelapan dikepalai oleh Masdarsaisa dengan masa periode Tahun 1995 sampai dengan 2000. Bupati Tanah Datar kesembilan di pimpin oleh Drs. Masriadi Martunus dengan masa periode Tahun 2000 sampai dengan 2005. Bupati Tanah Datar kesepuluh di kepalai oleh Shadiq Pasadigoe dengan masa jabatan 2005 sampai dengan 2015.

Kabupaten Tanah Datar berdaa di Batusangkar, uniknya Kota Batusangkar ini berda pada tiga (3) wilayah Kecamatan, yaitu Kecamatan Lima Kaum, Kecamatan tanjung Emas, dan Kecamatan Sungai Tarab. Sedangkan pusat pemerintahan berada di Kecamatan Tanjung Emas atau Tepanya di Nagari Pagaruyung. Kota Batusangkar ini lebih dikenal dengan Kota Budaya, karena di Kabupaten Tanah Datar terdapat banyak peninggalan dan prasasti terutama peninggalan Istana Basa Pagaruyung yang merupakan pusat Kerajaan Minangkabau.

Secara geografis wilayah Kabupaten Tanah Datar berada di sekitar kaki gunung Merapi, gunung Singgalang, dan gunung sago dan dipekaya dengan 25 sungai. Danau singkarak yang cukup luas sebagian diantaranya merupakan baian wilayah Kabupaten Tanah Datar yakni terletak di Kecamatan Batipuh Selatan dan Rambatan.

Diantara seluruh Kecamatan yang ada, 3 Kecamatan terletak pada ketinggian antara 750 s.d. 1000 meter di atas permukaan laut, yaitu Kecamatan X Koto, Salimpaung, dan Tanjung Baru. Sementara itu empat Kecamatan lainnya, yaitu Kecamatan Lima Kaum, Tanjung Emas, Padang Ganting, dan Sungai Tarab terletak pada ketinggian 450 s.d. 500 meter di permukaan laut. Sedangkan 7 Kecamatan lagi terletak pada ketinggian yang bervariasi, misalnya Kecamatan Lintau Buo yang terletak pada ketinggian antara 200 s.d. 700 meter dari permukaan laut.

2. Uraian Tugas Pokok Dan Fungsi Bagian Pemerintahan Nagari Dan Urusan Rantau Kabupaten Tanah Datar

Bagian Pemerintahan Nagari dan Urusan Rantau mempunyai tugas pokok menyiapkan perumusan kebijakan dan pengkoordinasian pelaksanaan kegiatan dibidang pemberdayaan kelembagaan nagari, pembinaan adminstrasi nagari, pemberdayaan potensi nagari dan perantau sesuai urusan yang menjadi kewenangan sebagaimana terdapat pada

lampiran peraturan bupati ini dan urusan lain untuk menunjang kebijakan bupati dalam penyelenggaraan pemerintahan.

Bagian Pemerintahan Nagari dan Urusan Rantau mempunyai fungsi sebagai berikut :

- a. Perumusan kebijakan umum pemerintah daerah bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;
- b. Penyusunan perencanaan strategis bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;
- c. Pengkoordinasian penyusunan bahan perumusan kebijakan bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;
- d. Menyiapkan bahan kebijakan pelaksanaan urusan yang menjadi kewenangan;
- e. Menyusun rencana kegiatan tahunan sesuai program dan urusan dengan mempedomani peraturan perundang-undangan serta kebijakan teknis lembaga pemerintah terkait
- f. Penyelenggaraan dan pelayanan administrasi bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;
- g. Pelaksanaan fasilitasi dan koordinasi penyelenggaraan dan pembinaan bidang pemberdayaan kelembagaan nagari, administrasi nagari, pemberdayaan potensi nagari dan perantau;
- h. Pengkoordinasian penyelenggaraan sistem informasi bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;
- i. Monitoring, evaluasi dan pelaporan bidang pemberdayaan kelembagaan nagari, pembinaan administrasi nagari, pemberdayaan potensi nagari dan perantau;

- j. Melaksanakan tugas penunjang dan tugas yang bersifat rutinitas sesuai kewenangan;
- k. Pelaksanaan tugas lain yang diberikan pimpinan sesuai dengan bidang tugas.

Bagian Pemerintahan Nagari dan Urusan Rantau terdiri dari:

- a. Subbagian Pemberdayaan Kelembagaan Nagari.
- b. Subbagian Pembinaan Adminstrasi Nagari.
- c. Subbagian Pemberdayaan Potensi Nagari dan Perantau.

Subbagian dipimpin oleh seorang Kepala Subbagian yang berada dibawah dan bertanggung jawab kepada Kepala Bagian.

3. Tugas Dan Tanggung Jawab Masing-Masing Sub Bagian

- a. Subbagian Pemberdayaan Kelembagaan Nagari

Subbagian Pemberdayaan Kelembagaan Nagari mempunyai tugas menyiapkan bahan penetapan kebijakan dan pengkoordinasian pelaksanaan dan pembinaan kegiatan ruang lingkup pemberdayaan kelembagaan nagari, sesuai urusan yang menjadi kewenangan bagian sebagaimana terdapat pada lampiran peraturan bupati ini;

Uraian Tugas Subbagian Pemberdayaan Kelembagaan Nagari :

- 1) Menghimpun peraturan perundang-undangan, pedoman, petunjuk teknis, data dan informasi, serta bahan lainnya yang berhubungan dengan bidang pemberdayaan kelembagaan nagari, sebagai pedoman dan landasan kerja;
- 2) Menginventarisir permasalahan yang berhubungan dengan bidang pemberdayaan kelembagaan nagari dan menyiapkan bahan petunjuk pemecahan masalah;
- 3) Menyiapkan bahan kebijakan pelaksanaan urusan yang menjadi kewenangan;

- 4) Menyusun rencana kegiatan tahunan sesuai program dan urusan dengan mempedomani peraturan perundang-undangan serta kebijakan teknis lembaga pemerintah terkait;
- 5) Mengkoordinasikan dan memfasilitasi pembinaan pemberdayaan kelembagaan nagari;
- 6) Monitoring, evaluasi dan pelaporan bidang pemberdayaan kelembagaan nagari;
- 7) Melaksanakan penyimpanan berkas kerja, data dan bahan menurut peraturan perundang-undangan;
- 8) Menyusun laporan pelaksanaan tugas dan kegiatan yang telah ditetapkan;
- 9) Mempertanggungjawabkan kegiatan yang telah dilaksanakan sesuai peraturan perundang-undangan.

b. Subbagian Pembinaan Administrasi Nagari

Subbagian Pembinaan Administrasi Nagari mempunyai tugas menyiapkan bahan penetapan kebijakan dan pengkoordinasian pelaksanaan dan pembinaan kegiatan ruang lingkup pembinaan administrasi nagari, sesuai urusan yang menjadi kewenangan bagian sebagaimana terdapat pada lampiran peraturan bupati ini;

Uraian tugas subbagian Pembinaan Administrasi Nagari :

- 1) Menghimpun peraturan perundang-undangan, pedoman, petunjuk teknis, data dan informasi, serta bahan lainnya yang berhubungan dengan bidang pembinaan administrasi nagari, sebagai pedoman dan landasan kerja;
- 2) Menginventarisir permasalahan yang berhubungan dengan bidang pembinaan administrasi nagari dan menyiapkan bahan petunjuk pemecahan masalah;
- 3) Menyiapkan bahan kebijakan pelaksanaan urusan yang menjadi kewenangan;

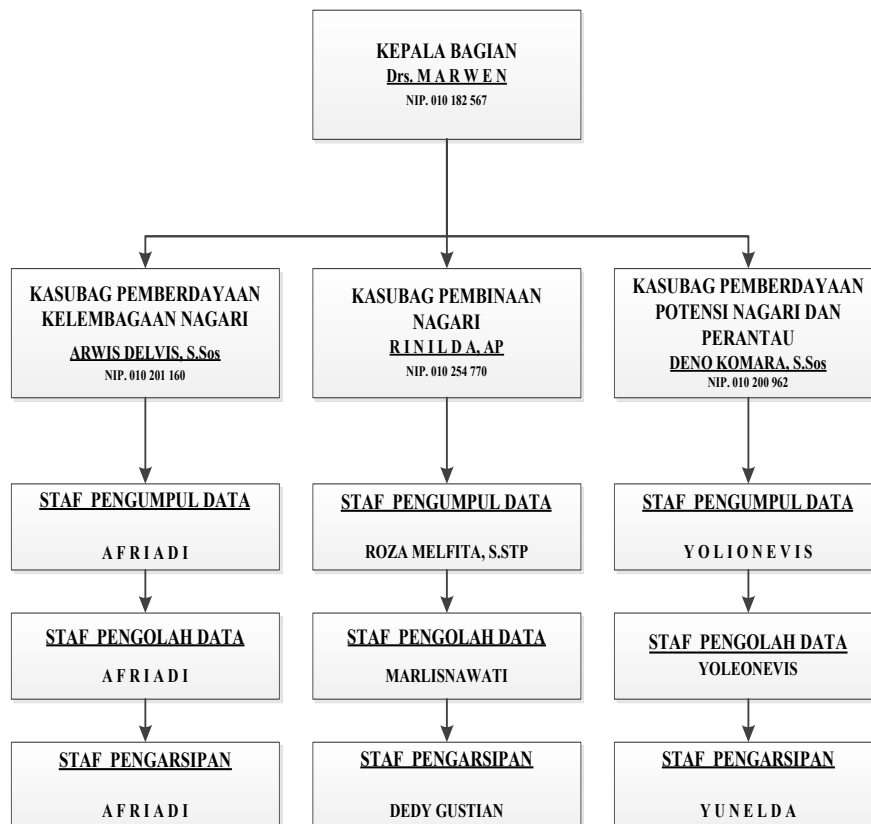
- 4) Menyusun rencana kegiatan tahunan sesuai program dan urusan dengan mempedomani peraturan perundang-undangan serta kebijakan teknis lembaga pemerintah terkait;
 - 5) Menyiapkan pedoman pelaksanaan tugas dan kegiatan;
 - 6) Melaksanakan dan mengkoordinasikan kegiatan dengan unit kerja terkait;
 - 7) Mengkordinir pelaksanaan kegiatan bidang pembinaan administrasi nagari;
 - 8) Mengkoordinasikan dan memfasilitasi pembinaan bidang administrasi nagari;
 - 9) Monitoring, evaluasi dan pelaporan bidang pembinaan administrasi nagari;
 - 10) Melaksanakan penyimpanan berkas kerja, data dan bahan menurut peraturan perundang-undangan;
 - 11) Menyusun laporan pelaksanaan tugas dan kegiatan yang telah ditetapkan;
 - 12) Mempertanggungjawabkan kegiatan yang telah dilaksanakan sesuai peraturan perundang-undangan;
 - 13) Melaksanakan tugas penunjang dan tugas yang bersifat rutinitas sesuai kewenangan;
 - 14) Melaksanakan tugas lain yang diberikan oleh pimpinan sesuai dengan bidang tugasnya.
- c. Subbagian Pemberdayaan Potensi Nagari dan Perantau

Subbagian Pemberdayaan Potensi Nagari dan Perantau mempunyai tugas menyiapkan bahan penetapan kebijakan dan pengkoordinasian pelaksanaan dan pembinaan kegiatan ruang lingkup pemberdayaan potensi nagari dan perantau, sesuai urusan yang menjadi kewenangan bagian sebagaimana terdapat pada lampiran peraturan bupati ini;

Uraian Tugas Bagian Pemberdayaan Potensi Nagari dan Perantau :

- 1) Menghimpun peraturan perundang-undangan, pedoman, petunjuk teknis, data dan informasi, serta bahan lainnya yang berhubungan dengan bidang pemberdayaan potensi nagari dan perantau, sebagai pedoman dan landasan kerja;
- 2) Menginventarisir permasalahan yang berhubungan dengan bidang pemberdayaan potensi nagari dan perantau serta menyiapkan bahan petunjuk pemecahan masalah;
- 3) Menyiapkan bahan kebijakan pelaksanaan urusan yang menjadi kewenangan;
- 4) Menyusun rencana kegiatan tahunan sesuai program dan urusan dengan mempedomani peraturan perundang-undangan serta kebijakan teknis lembaga pemerintah terkait;
- 5) Menyiapkan pedoman pelaksanaan tugas dan kegiatan;
- 6) Melaksanakan dan mengkoordinasikan kegiatan dengan unit kerja terkait;
- 7) Mengkordinir pelaksanaan kegiatan bidang pemberdayaan potensi nagari dan perantau;
- 8) Mengkoordinasikan dan memfasilitasi pembinaan bidang pemberdayaan potensi nagari dan perantau;
- 9) Monitoring, evaluasi dan pelaporan bidang pemberdayaan potensi nagari dan perantau;
- 10) Melaksanakan penyimpanan berkas kerja, data dan bahan menurut peraturan perundang-undangan;
- 11) Menyusun laporan pelaksanaan tugas dan kegiatan yang telah ditetapkan;
- 12) Mempertanggungjawabkan kegiatan yang telah dilaksanakan sesuai peraturan perundang-undangan;
- 13) Melaksanakan tugas penunjang dan tugas yang bersifat rutinitas sesuai kewenangan;
- 14) Melaksanakan tugas lain yang diberikan oleh pimpinan sesuai dengan bidang tugasnya.

4. Struktur Organisasi Pemerintahan Nagari dan Urusan Rantau



Gambar 2.1 Struktur Organisasi Pemerintahan Nagari Dan Urusan Rantau)

5. Visi dan Misi

a. Visi

Visi merupakan cara pandangan yang jauh kedepan kemana dan bagaimana Pemerintahan Kabupaten Tanah Datar harus dibawa dan bekarya agar tetap konsisten dan eksis, antisipatif, inovatif, serta produktif.

Visi adalah gambaran umum yang menantang tentang keadaan masa depan berisikan cita dan citra yang ingin diwujudkan oleh instansi pemerintah.

Pemerintah Kabupaten Tanah Datar mempunyai Visi: “MENJADIKAN PEMERINTAHAN KABUPATEN TANAH DATAR MAJU DALAM SEGALA BIDANG DAN BERBUDAYA

YANG BERLANDASKAN ADAT BESANDI SYARAK, SYARAK BASANDI KITABULLAH.

b. Misi

Misi merupakan kegiatan utama yang hendak dilakukan yang membuat organisasi memiliki jati diri yang jelas dan sekaligus membedakannya dari organisasi lain. Misi adalah sesuatu yang harus dilaksanakan oleh Pemerintahan Kabupaten Tanah Datar sesuai visi yang ditetapkan, agar tujuan organisasi dapat terlaksana dan berhasil dengan baik.

Berdasarkan visi yang telah ditetapkan maka Pemerintahan Kabupaten Tanah Datar menetapkan misi sebagai berikut :

- 1) Meningkatkan keimanan dan ketaqwaan masyarakat Kabupaten Tanah Datar serta berakhlak dan berbudaya.
- 2) Mengadaan pembinaan bagi anak yang putus sekolah yang belum mempunyai keterampilan.
- 3) Meningkatkan pelayanan yang baik kepada masyarakat.
- 4) Meningkatkan partisipasi masyarakat.
- 5) Meningkatkan secara maksimal pembinaan keagamaan dan sosial budaya.
- 6) Menjaga keamanan dan ketertiban masyarakat serta mengadakan penyuluhan hukum dan pekat.
- 7) Melaksanakan tata cara pemerintahan yang baik dan beribawa.

B. Konsep Dasar Sistem Informasi

1. Sistem

a. Pengertian Sistem

Kata sistem dalam bahasa inggris berarti *system*. Menurut Hassan (2003) Dalam Kamus Inggris-Indonesia, *System* berarti susunan atau cara. Sedangkan menurut Fathansyah (2007) menyatakan bahwa suatu sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi atau tugas

khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu.

Menurut Jogiyanto (2005) menyatakan bahwa sistem dapat didefinisikan dengan dua kelompok pendekatan yaitu pendekatan prosedur dan pendekatan komponen. Sistem berdasarkan pendekatan prosedur adalah suatu kumpulan prosedur-prosedur yang saling berhubungan untuk melakukan suatu kegiatan. Sistem berdasarkan pendekatan komponen atau elemen adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Sistem menurut Wahyono (2004) adalah sistem dapat terdiri atas kegiatan-kegiatan yang berhubungan guna mencapai tujuan-tujuan perusahaan seperti inventaris atau penjadwalan produksi. Sedangkan menurut penulis sistem itu adalah elemen-elemen yang saling berhubungan untuk mencapai tujuan dari sistem itu sendiri.

Dari beberapa pengertian sistem di atas dapat disimpulkan bahwa sistem adalah kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem yang saling berhubungan membentuk suatu kesatuan hingga tujuan atau sasaran tersebut tercapai.

b. Syarat-syarat Sistem

Menurut Faisal (2008) syarat-syarat sistem adalah sebagai berikut:

- 1) Sistem harus dibentuk untuk menyelesaikan tujuan.
- 2) Elemen sistem harus mempunyai rencana yang ditetapkan.
- 3) Adanya hubungan diantara elemen sistem.
- 4) Unsur dasar dari proses (arus informasi, energi dan material) lebih penting daripada sistem.
- 5) Tujuan organisasi lebih penting daripada tujuan elemen.

c. Pengelompokan Sistem

Menurut Faisal (2008) Sistem dikelompokkan bertujuan untuk mempermudah mengetahui sistem apa yang akan dibangun. Sistem

yang telah dikelompokkan akan membantu pengguna dalam mengoperasikan sistem tersebut.

2. Informasi

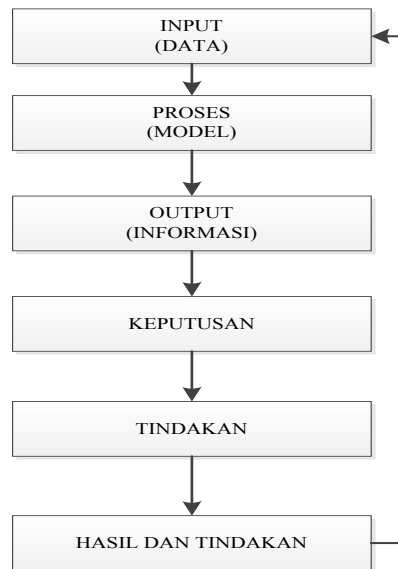
a. Pengertian Informasi

Informasi dalam bahasa Inggris berarti *Information*. Menurut Hassan (2003) *Information* berarti: penerangan, pemberitahuan. Menurut Faisal (2008) Informasi adalah data yang telah diolah dan siap digunakan oleh pengambil keputusan. Informasi merupakan produk akhir dari suatu sistem. Nilai suatu informasi berhubungan dengan pengambilan keputusan. Dalam penetapan nilai. Pengambilan keputusan tentunya mempelajari, menelaah informasi tersebut apakah layak untuk dilempar ke pasaran atau tidak dengan pemberian nilai jual terhadap informasi tersebut.

Sedangkan menurut Jogiyanto (2005) Informasi dapat di definisikan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Informasi suatu yang berharga karena dengan penguasaan yang baik atas informasi tersebut perusahaan akan dapat mengambil atau memperoleh kesempatan yang lebih luas dan berkembang. Dengan adanya informasi yang memadai suatu perusahaan akan dapat mengambil keputusan yang akan mendukung kemajuan perusahaan itu sendiri.

Informasi menurut Kusrini dan Koniyo (2007) adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai apabila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

Dari definisi diatas dapat disimpulkan bahwa sumber informasi adalah data. Data belum menunjukkan sesuatu yang bisa dipahami, karenanya harus diproses terlebih dahulu.



Gambar 2.2 Siklus Informasi (Jogiyanto, 2005)

b. Pengguna Informasi

Menurut Faisal (2008) pengguna informasi adalah orang atau badan yang mempunyai kepentingan terhadap informasi yang dipakai. Pengguna informasi dalam bidang manajemen adalah manajer dan anak buahnya. Pengguna informasi dalam bidang pendidikan adalah para pendidik dan anak didiknya. Pengguna informasi dalam bidang telekomunikasi adalah para manajer dan anak buahnya. Pengguna informasi dalam bidang pertahanan adalah para komandan dan prajuritnya. Pengguna informasi dalam bidang teknologi informasi adalah para programmer, *end-user*, administrator jaringan, *technical support*, *graphic designer*.

3. Sistem Informasi

a. Pengertian Sistem Informasi

Menurut Faisal (2008) Sistem Informasi dibuat untuk menghasilkan informasi yang berkualitas. Sistem Informasi adalah

suatu sistem manajemen organisasi baik secara manajerial maupun secara strategis dalam menyediakan laporan yang diperlukan kepada pihak luar.

Sistem Informasi menurut Kusrini dan Koniyo (2007) adalah sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

Sedangkan Robert dan Bavis (2002) menyatakan Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut Faisal (2008) semua Sistem Informasi memiliki kegiatan utama, yaitu :

- 1) Menerima data sebagai masukan (input)
- 2) Melakukan pemrosesan dengan mengerjakan perhitungan, penggolongan, unsur data, dan pemutakhiran (updating)
- 3) Memperoleh informasi sebagai keluaran (output)

Dengan demikian, Sistem Informasi berdasarkan konsep (input-proses output) dapat dilihat pada gambar berikut:



Gambar 2.3 Konsep Sistem Informasi (Faisal, 2008)

b. Komponen Sistem Informasi

Menurut Kusrini dan Andri Koniyo (2007) yang termasuk komponen sistem informasi adalah :

- 1) Perangkat keras (*hardware*), mencakup berbagai piranti fisik seperti *Personal Computer* dan *Printer*.

- 2) Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
- 3) Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
- 4) Orang (*user*), yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
- 5) Basis data (*database*), yaitu sekumpulan tabel, hubungan dan lain-lain yang terkait dengan penyimpanan data.
- 6) Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resource*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

4. Perancangan Sistem Informasi

a. Pengertian Perancangan Sistem

Nugroho (2005) menyatakan bahwa perancangan sistem adalah tahap awal dimana pendekatan awal untuk menyelesaikan/memecahkan masalah dan mengembangkan solusi terbaik bagi permasalahan itu. Sedangkan menurut Jogiyanto (2005) Perancangan Sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan sistem yang utuh dan berfungsi.

Jadi, perancangan sistem merupakan cara bagaimana mengorganisasikan sistem ke dalam subsistem-subsistem, serta alokasi subsistem-subsistem ke komponen-komponen perangkat keras, perangkat lunak, serta prosedur-prosedur.

b. Tujuan Perancangan Sistem

Menurut Jogiyanto (2005) tujuan yang akan dicapai dalam perancangan suatu sistem adalah:

- 1) Perancangan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
- 2) Untuk memenuhi kebutuhan kepada pemakai sistem.
- 3) Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan data transaksi manajemen dan mendukung keputusan yang diambil oleh pihak manajemen.
- 4) Perancangan sistem harus dapat mendukung tujuan utama perusahaan.
- 5) Perancangan sistem harus dapat mempersiapkan rancangan bangunan yang terinci untuk masing-masing komponen dari sistem informasi.

C. Alat Bantu Perancangan Sistem

Untuk dapat melakukan langkah-langkah pengembangan sistem sesuai dengan metodologi pengembangan sistem yang terstruktur, maka dibutuhkan alat dan teknik untuk melaksanakannya. Alat-alat yang digunakan dalam suatu perancangan system umumnya berupa gambar dan diagram.

Adapun alat bantu yang digunakan dalam perancangan sistem yang akan digunakan dalam penulisan adalah:










1. Aliran Sistem Informasi (ASI)

Menurut Jogiyanto (2005) Aliran sistem informasi disebut juga bagan alir (*flowchart*). Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur secara logika.

ASI merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada didalam sistem. ASI menunjukkan apa yang dikerjakan di sistem.

Simbol-simbol yang umum digunakan sebagai berikut :

Tabel 2.1. Simbol Aliran Sistem Informasi (Jogiyanto, 2005)



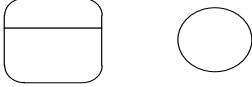

No	Simbol	Keterangan
1		Proses Komputerisasi
2		Proses manual
3		Dokumen
4		Penyimpanan
5		Hardisk
6		menggunakan Input menggunakan keyboard
7		Arus data
8		Menunjukkan output yang ditampilkan pada monitor
9		Menunjukkan penjelasan Terhadap suatu proses

2. Data Flow Diagram (DFD)

Menurut Jogiyanto (2005) diagram arus data (*DFD*) merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*Structured Analysis and design*). DFD dapat menggambarkan arus data di dalam sistem dengan struktur yang jelas. Lebih lanjut DFD juga merupakan dokumentasi dari sistem yang baik.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. DFD merupakan gambaran sistem secara logikal. Namun gambar tersebut tidak tergantung pada perangkat keras, perangkat lunak, struktur data atau organisasi file. Keuntungan DFD adalah memudahkan pemakai yang kurang menguasai bidang komputer untuk menguasai sistem yang akan dikerjakan atau dikembangkan. Adapun simbol-simbol yang digunakan dalam DFD adalah:

Tabel 2.2. Simbol Data Flow Diagram (DFD) (Jogiyanto, 2005)

No	Simbol	Keterangan
1		Kesatuan Luar (<i>Eksternal Entity</i>)
2		Arus Data (<i>Data Flow</i>)
3		Proses (<i>Process</i>)
4		Penyimpanan (<i>Data Store</i>)


Menurut Jogiyanto (2005) aturan umum dalam penggambaran DFD adalah :


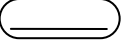
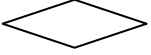
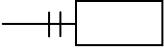
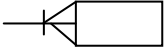
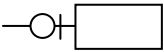
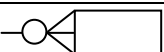
- a) Tidak boleh menghubungkan eksternal *entity* dengan eksternal *entity* lainnya secara langsung
- b) Tidak boleh menghubungkan secara langsung antara data store dengan data store lainnya
- c) Tidak boleh menghubungkan data store dengan eksternal *entity* secara langsung
- d) Suatu proses harus menerima arus data dan menghasilkan arus data
- e) Proses harus mempunyai nama dan nomor.

3. Entity Relationship Diagram (ERD)

Menurut Nugroho (2004) *Entity Relationship diagram* (ERD) adalah diagram yang memperlihatkan entitas-entitas yang terlibat dalam suatu sistem serta hubungan-hubungan (*relation*) antar entitas tersebut. Simbol-simbol yang digunakan dalam *Entity Relationship Diagram* sebagai berikut :

Tabel 2.3 Simbol Entity Relationship Diagram (Nugroho, 2004)

No	Simbol	Arti/Tujuan
1		Entity

2		Atribut dari entity
3		Atribut dari entity dengan key
4		Relasi antar entity
5		Hubungan satu dan pasti
6		Hubungan banyak dan pasti
7		Hubungan satu tapi tidak pasti
8		Hubungan satu tapi tidak pasti

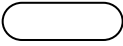
Menurut Nugroho (2004) Bagian dari Entity Relationship Diagram (ERD) adalah:

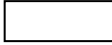
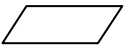
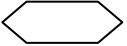
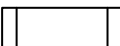
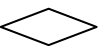

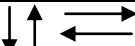
- a) *Entity* adalah objek yang dapat didefinisikan secara unik dengan objek lainnya.
- b) *Attribut* yaitu karakteristik dari *entity* yang menyediakan penjelasan detail tentang *entity* tersebut
- c) *Relationship* atau hubungan yang terjadi antara satu *entity* dengan *entity* lainnya.

4. Program Flowchart

Program Flowchart atau Bagan Alir Program menurut Jogiyanto (2005) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. *Program Flowchart* dibuat dengan menggunakan simbol-simbol sebagai berikut:

Tabel 2.4 Simbol Program Flowchart (Jogiyanto, 2005)

No	Simbol	Arti/Tujuan
1		Terminal, menunjukkan awal dan akhir suatu proses

2		Simbol proses, digunakan untuk mewakili suatu proses
3		input atau output, digunakan untuk mewakili data i/o
4		Simbol persiapan digunakan untuk memberi nilai awal suatu besaran
5		Proses terdefenisi, menunjukan suatu operasi yang rinciannya ditunjukkan di tempat lain
6		Keputusan, digunakan untuk suatu seleksi kondisi di dalam program
7		Penghubung, menunjukan penghubung ke halaman yang sama atau ke halaman lain
8		Garis air, menunjukan arus dari proses

5. Normalisasi

Menurut Jogiyanto (2005) normalisasi adalah proses untuk mengorganisasikan *file* untuk menghilangkan grup elemen yang berulang-ulang. Proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang ini disebut dengan normalisasi (*Normalization*). Sedangkan menurut Prayudi (2006) Normalisasi adalah suatu proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam dua buah tabel atau lebih, yang tidak lagi memiliki masalah tersebut. Masalah tersebut biasanya merupakan suatu ketidakkonsistenan

(tidak normal) apabila dilakukan penghapusan (*delete*), perubahan (*update*) dan pembacaan (*retrieve*) pada suatu basis data.

Menurut Simarmata dan Prayudi (2006) normalisasi adalah Bagian perancangan basis data yang berkaitan dengan model data relational untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Sebuah Basisdata dapat dikatakan baik, jika setiap tabel yang menjadi unsur pembentuk basisdata tersebut telah berada dalam keadaan baik atau normal.

Dari definisi diatas, dapat disimpulkan bahwa: normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang mengandung redundansi sesedikit mungkin, memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Bentuk Normalisasi adalah suatu aturan yang dikenakan pada tabel-tabel dalam basis data dan harus dipenuhi oleh tabel-tabel tersebut pada level-level normalisasi. Ada macam-macam bentuk normalisasi, diantaranya adalah bentuk tidak normal, bentuk normal pertama, bentuk normal kedua dan bentuk normal ketiga.

Menurut Prayudi (2006) Aturan-aturan dalam masing-masing bentuk normalisasi tersebut adalah sebagai berikut:

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

b. Bentuk normal pertama

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom (tidak perlu ada indeks dalam memberi nama kolom).

c. Bentuk normal kedua

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

d. Bentuk normal ketiga

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

D. Database

1. Pengertian

Menurut Wiljayanto (2000) Database merupakan kumpulan data tentang suatu benda atau kejadian yang saling berhubungan antara yang satu dengan yang lainnya. Konsep *Database* merupakan suatu integrasi logis dari catatan-catatan *file*. Tujuan dari *database* ini adalah meminimumkan pengulangan dan mencapai independensi data program. Program mengacu pada table untuk mengakses data.

Jika kita ingin membuat suatu program manajemen *database*, kita memerlukan *file database* yang akan diproses. *File database* dibuat hanya satu kali dan tidak perlu langsung diisi recordnya. Selanjutnya *file database* diproses lewat program. Untuk membuat *file database* digunakan program Visual Data Manager (VisData) atau program lain seperti *Microsoft Access 2010*, *Microsoft SQL Server 2008*.

Dalam merancang *database* perlu mempertimbangkan dua hal, yaitu:

- 1) Informasi apa yang akan disimpan sehingga kita dapat menentukan entitas yang harus disimpan informasinya.

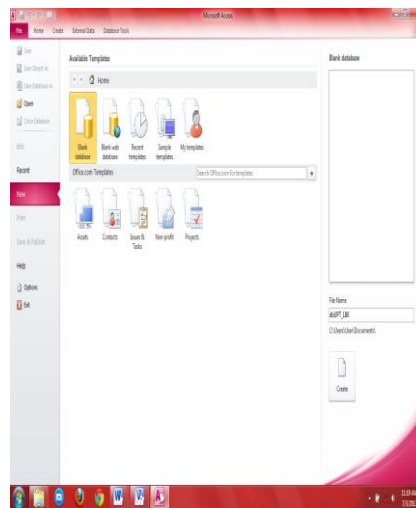
- 2) Pertanyaan yang akan diajukan terhadap *database* sering disebut *queries*.

E. Mengenal Elemen Dasar Microsoft Access 2010

Madcoms (2010) *Microsoft Access* merupakan program *database* yang cukup populer dan banyak digunakan saat ini. Ini dikarenakan *Microsoft Access* lebih mudah dalam mengolah berbagai jenis database serta hasil akhir yang berupa laporan dengan tampilan desain yang lebih menarik. Dalam *MS-Access* 2010, akan ditemukan tampilan yang sedikit berbeda dengan versi sebelumnya, dengan pengoperasian yang lebih mudah namun tidak mengubah fungsi dan keefektifannya.

1. Memulai Microsoft Access 2010

Setelah menjalankan *MS-Access* 2010, maka akan ditampilkan *Backstage View* seperti pada Gambar 2.4 *Backstage View* merupakan jendela awal dan fitur terbaru dari *MS-Access* 2010. Pada Jendela awal *Microsoft Access* 2010 ini, kita dapat membuka lembar kerja *MS-Access* dengan membuat atau membuka database.



Gambar 2.4 Jendela program Microsoft Access 2010 (Madcoms, 2010)

2. Komponen Microsoft Acces 2010

Menurut Madcoms (2010) Komponen-komponen yang terdapat dalam *MS-Access* 2010 adalah sebagai berikut ;

- a. *Title Bar*, berfungsi untuk menampilkan nama dari jendela program atau nama dari *file* yang aktif.
- b. *Status Bar*, bagian yang digunakan untuk menampilkan informasi dan kondisi dari lembar kerja program *MS-Access 2010*. Misalnya: kondisi status aktif *Caps Lock*, *Numb Lock*, *Scroll Lock* dan sebagainya.
- c. *Ribbon*, merupakan area di atas jendela program dimana anda dapat memilih tombol perintah.
- d. *Command Tab*, merupakan kumpulan dari grup yang berbentuk tabulasi secara otomatis menyesuaikan dokumen aktif.
- e. *Contextual Command Tab*, merupakan tab yang tampil secara otomatis berdasarkan dokumen atau objek yang sedang dikerjakan.
- f. *Group*, merupakan isi dari tab yang berupa kumpulan dari beberapa tombol perintah yang saling berkaitan.
- g. *Navigation Pane*, bagian pada sisi sebelah kiri jendela *database* yang menampilkan objek-objek *database*.

3. Mengenal Database *MS-Access 2010*

Menurut Madcoms (2010) *Database* merupakan sekumpulan data atau informasi yang terdiri atas tabel-tabel yang saling berhubungan antara satu dengan yang lainnya, dimana kita mempunyai wewenang untuk mengakses data tersebut, baik menambah, mengganti, menghapus, dan mengedit data dalam tabel-tabel tersebut. Data-data yang diolah tersebut disimpan dalam sebuah file dengan ekstensi *.accdb* (*Access Database*).

Berbeda dengan program *Microsoft Office* lain dalam pengoperasiannya, *MS-Access* hanya mampu digunakan untuk mengolah satu *file database*. Sedangkan *table* merupakan tempat untuk menyimpan data yang terdiri dari beberapa bagian berikut :

- a. *Field*, merupakan tempat data atau informasi dalam kelompok sejenis yang dimasukkan atau diinputkan pada bagian kolom *table*.
- b. *Record*, merupakan kumpulan dari beberapa *field* yang saling berhubungan dan tersimpan dalam bentuk baris pada *table*. Satu *table* dapat terdiri dari banyak *record* sekaligus

4. Tipe Data

Tipe data pada Microsoft Access 2010 dapat dilihat pada tabel 2.5 sebagai berikut:

Tabel 2.5 Tipe data Microsoft Access 2010 (Madcoms, 2010)

Tipe	Keterangan
<i>Text</i>	Untuk menerima data teks sampai 255 karakter yang terdiri dari huruf, angka, dan symbol grafik.
<i>Memo</i>	Untuk menerima data teks sampai 65.535 karakter yang terdiri dari huruf, bilangan, tanda baca, serta symbol grafik. Tipe data ini tidak dapat digunakan sebagai acuan untuk pengurutan data (<i>indeks</i>).
<i>Number</i>	Untuk menerima digit, tanda minus dan titik decimal. Tipe data number mempunyai pilihan ukuran bilangan dan jumlah dua digit tertentu.
<i>Date/Time</i>	Untuk menerima data tanggal dan waktu, serta nilai tahun yang dimulai dari tahun 100 sampai tahun 9999.
<i>Currency</i>	Untuk menerima data digit, tanda minus dan tanda titik dengan tingkat 15 digit desimal disebelah kiri tanda titik decimal dan 4 digit di sebelah kanan titik desimal.
<i>AutoNumber</i>	Untuk menampilkan nomor urut otomatis, yaitu berupa data angka mulai dari 1 dengan selisih

	1.
<i>Yes/No</i>	Tipe ini untuk menerima salah satu data dari dua nilai, yaitu <i>Yes/No</i> , <i>True/False</i> , atau <i>On/Off</i>
<i>OLE Objek</i>	Untuk menerima data yang berupa objek grafik, <i>spreadsheet</i> , foto digital, rekaman suara, atau video yang dapat diambil dari program aplikasi lain. Ukuran maksimum adalah 1 <i>gigabyte</i> .
<i>Hyperlink</i>	Untuk menerima data yang berupa teks yang berwarna dan bergaris bawah grafik, serta tipe data ini berhubungan dengan jaringan.
<i>Attachment</i>	Untuk menerima data yang berupa <i>file</i> gambar, <i>spreadsheet</i> , document, grafik, dan tipe <i>file</i> lainnya.
<i>Calculated</i>	Untuk menerima data berupa rumus perhitungan dari <i>field-filed</i> tertentu.
<i>Lookup Wizard</i>	Untuk menampilkan satu dari beberapa tipe data yang ada pada suatu daftar. Data tersebut dapat diambil dari table maupun <i>query</i> yang ada.

Menurut Madcoms (2010) untuk *Field Name* yang bertipe *Number* ketentuan panjang ukuran yang dapat dimasukkan ditentukan berdasarkan pilihan propertinya. *Integral Field Name* dengan bertipe *Number* dapat dilihat pada tabel 2.6 sebagai berikut :

Tabel 2.6 Tabel Integral Field Name dengan tipe Number (Madcoms, 2010)

Pilihan	Keterangan
<i>Byte</i>	Memberikan nilai integer (Bilangan Bulat) dari 0 sampai 225 dan tidak termasuk pecahan.
<i>Decimal</i>	Memberikan nilai dari -10E28 sampai 10E28 dengan ketelitian 28 digit dibelakang titik desimal.
<i>Integer</i>	Memberikan nilai <i>integer</i> (bilangan bulat) dari -32.768 sampai 32.768 dan tidak termasuk pecahan.
<i>Long Integer</i>	Memberikan nilai <i>integer</i> dari -2.147.483.648 sampai 2.147.483.648 dan tidak termasuk pecahan.
<i>Single</i>	Memberikan nilai bilangan nyata (bilangan yang mungkin memiliki pecahan terkecil) dengan ketelitian sampai 7 digit desimal, yang dimulai dari -3.402823E38 sampai -1.401298E45 untuk nilai negatif dan 1.401298E45 sampai 3.402823E38 untuk nilai positif.
<i>Double</i>	Memberikan nilai bilangan nyata (bilangan yang mungkin memiliki pecahan terkecil) dengan ketelitian sampai 15 digit dibelakang titik desimal, yang dimulai dari -1.7200069313486231E308 sampai dengan -4.94065645841247E-324 untuk nilai negative dan 1.7200069313486231E308 sampai 4.94065645841247E-324 untuk nilai positif.

F. Tinjauan Tentang Visual Basic 2010

1. Sejarah

Menurut Kurniawan (2010) Sejarah Visual Basic 2010 berawal dari Visual Basic yang diturunkan dari bahasa BASIC. Visual Basic terkenal sebagai bahasa pemrograman yang mudah untuk digunakan terutama untuk membuat aplikasi yang berjalan diatas *platform* Windows.

Pada tahun 90an, Visual menjadi bahasa pemrograman yang saling populer dan menjadi pilihan utama untuk mengembangkan program berbasis Windows. Versi Visual Basic terakhir sebelum berjalan diatas .NET Framework adalah VB6 (Visual Studio 1998).

Visual Basic .NET dirilis pada bulan Februari tahun 2002 bersamaan dengan *platform* .NET, yaitu VB 2002 (VB7), VB 2005 (VB8), VB 2008 (VB9), dan yang terakhir adalah VB 2010 (VB10) yang dirilis bersamaan dengan *Visual Studio* 2010.

Selain *Visual Basic* 2010, *Visual Studio* 2010 juga mendukung beberapa bahasa lain, yaitu C#, C++, F# (bahasa baru untuk *functional programming*), IronPhyton, dan IronRuby (bahasa baru untuk *dynamic programming*).

2. Memulai Visual Basic 2010

Pada saat *Visual Basic* 2010 pertama kali dijalankan sesudah proses instalasi berhasil, akan tampil halaman untuk memilih salah satu dari beberapa pengaturan yang disediakan.

Jika sudah terbiasa menggunakan VB6 atau VB.NET pada *Visual Studio* versi sebelumnya maka disarankan untuk memilih *Visual Basic Development Setting* karena pengaturannya sudah disesuaikan dengan kebiasaan programmer *Visual Basic*.

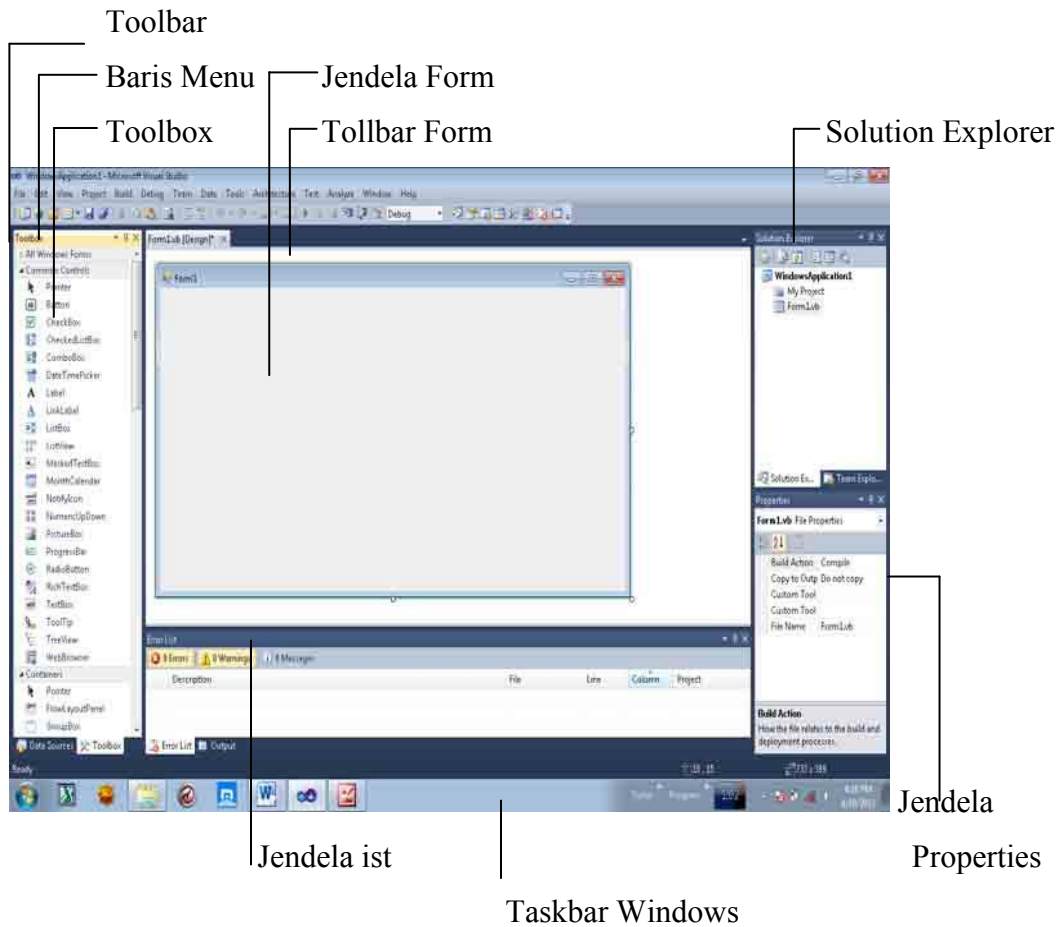
Tampilan *Visual Basic* 2010 pada saat pertama kali dijalankan adalah seperti pada gambar dibawah ini:



Gambar 2.5 Halaman StartUp (Kurniawan, 2010)

3. Komponen Visual Basic 2010

Menurut Kurniawan (2010) Komponen-komponen Pemrograman *Visual Basic 2010* berisi semua alat bantu yang diperlukan untuk membuat program-program.. Tampilan pertama saat *Visual Basic* dijalankan dapat dilihat pada gambar 2.6

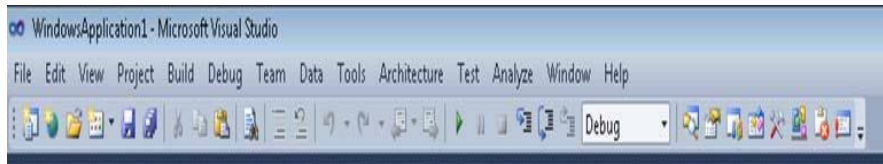


Gambar 2.6 Tampilan Pertama Visual Basic 2010 (Kurniawan, 2010)

a. Jendela Utama (Main Windows)

Menurut Kurniawan (2010) Terdiri dari *title bar* (baris judul), *menu bar*, dan *toolbar*. Baris judul berisi nama proyek, mode operasi *Visual Basic 2010* dan *form* yang aktif. *Menu bar* merupakan menu *drop-down* yang dapat mengontrol operasi dari lingkungan *Visual Basic*.

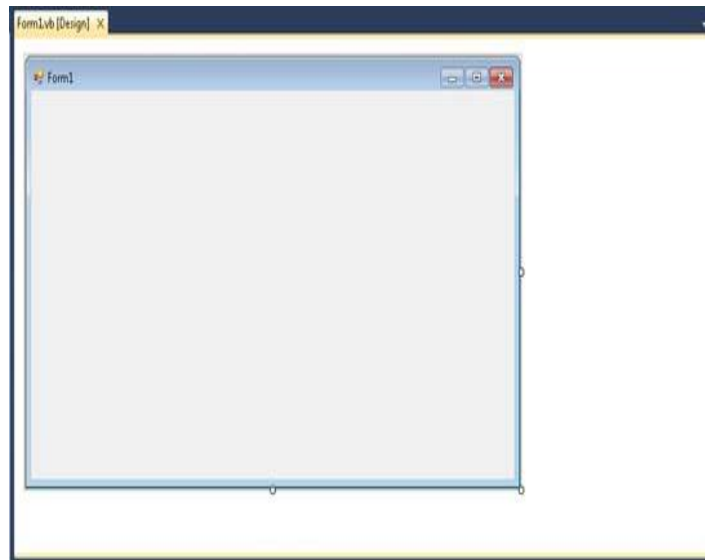
Toolbar berisi kumpulan gambar yang mewakili perintah yang ada di menu. Jendela utama juga menampilkan lokasi dari *form* yang relatif aktif terhadap sudut kiri atas layar (satuan ukurannya *twips*), juga lebar dan panjang dari *form* yang aktif. Gambar dari jendela utama dapat dilihat pada gambar 2.7



Gambar 2.7 Main Windows (Kurniawan, 2010)

b. Jendela Form (*Form Windows*)

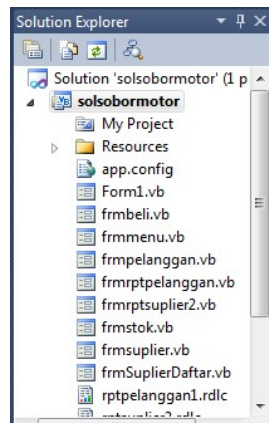
Menurut Kurniawan (2010) Jendela Form adalah pusat dari pengembangan aplikasi *Visual Basic 2010*. Pada *form* ini akan diletakkan berbagai macam objek interaktif seperti teks, gambar, tombol-tombol perintah, *scrollbar*, dan sebagainya. Gambar *Jendela Form* dapat dilihat pada gambar 2.8



Gambar 2.8 Form Windows (Kurniawan, 2010)

c. Project Windows (*Solution Explorer*)

Menurut Kurniawan (2010) *Project Windows* adalah jendela yang menampilkan daftar *form* dan modul project. Project merupakan kumpulan dari *modul form*, *modul class*, *modul standar*, dan *file* sumber yang membentuk suatu aplikasi. Dapat dilihat pada gambar 2.9

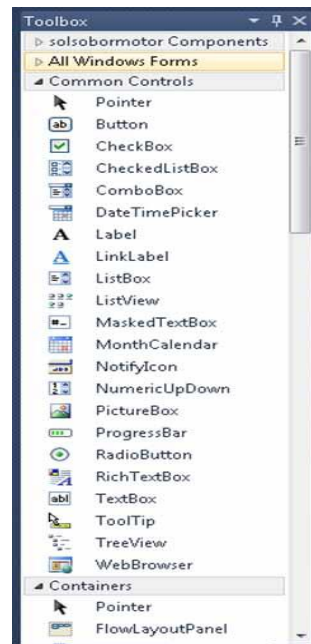


Gambar 2.9 Solution Explorer (Kurniawan, 2010)

d. Toolbox

Menurut Kurniawan (2010) *Toolbox* adalah sebuah kotak piranti yang mengandung semua objek atau kontrol yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi *interface* antara program aplikasi dengan user-nya, dan kesemuanya harus diletakkan didalam jendela *form*. Seperti terlihat pada gambar 2.10 halaman berikut:

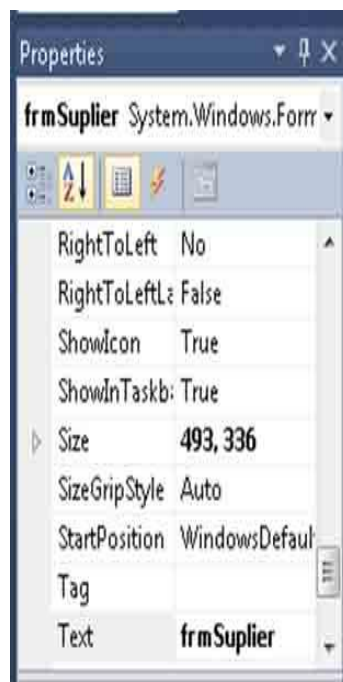
Sumber: Kurniawan (2010)



Gambar 2.10 Toolbox (Kurniawan, 2010)

e. Jendela Properti (*Properties Windows*)

Menurut Kurniawan (2010) Jendela Properti (*Properties Windows*) Berisi daftar struktur *setting properties* yang digunakan pada sebuah objek terpilih. Kotak *drop-down* pada bagian atas jendela berisi daftar semua objek pada form yang aktif. Ada dua tab tampilan: *Alphabetic* (urut abjad) dan *Categorized* (urut berdasarkan kelompok). Di bagian bawah kotak terdapat *properties* dari objek terpilih. Berikut gambar 2.11 dari jendela *properties*:



Gambar 2.11 Jendela *Properties* (Kurniawan, 2010)

f. Jendela Kode (*Code Windows*)

Menurut Kurniawan (2010) jendela kode (*Code Windows*) adalah salah satu jendela yang penting didalam *Visual Basic*. Jendela ini berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi *Visual Basic 2010*. Jendela Kode (*Code Windows*) dapat dilihat pada gambar 2.12

```

frmSuplier.vb - X  frmSuplier.vb [Design]  frmRptSuplier.vb [Design]  rptSuplier.rpt [Design]  frmBekub.vb  frmBekub [Design]
frmSuplier
simpanData

'jika data sudah ada
Dim con As New System.Data.OleDb.OleDbConnection
Dim cmd As System.Data.OleDb.OleDbCommand
Dim dr As System.Data.OleDb.OleDbDataReader
Dim strSQL As String

con.ConnectionString = "provider=microsoft.ace.oledb.12.0;data source=" & _
Application.StartupPath & "\sbornotor.accdb"
con.Open()
cmd = New OleDb.OleDbCommand("select * from tbsupplier where Namasupplier=" & txtNamasupplier.Text & """, con)
cmd = New OleDb.OleDbCommand("select * from tbsupplier where Kodesupplier=" & txtKodesupplier.Text & """, con)
dr = cmd.ExecuteReader
If dr.Read Then
    MsgBox(dr.Item("Kodesupplier"))
    MsgBox(dr.Item("Namasupplier"))
    'MsgBox(dr.Item("namasupplier"))
    "-jika perbaikan-ya
    If MsgBox("Data " & dr.Item("Namasupplier") & " sudah ada apakah ingin dihapus?", MsgBoxStyle.YesNo + MsgBoxStyle.Question) = MsgBoxStyle.Yes Then
        'begin perintah delete
        strSQL = "delete * from tbsupplier where Kodesupplier=" & txtKodesupplier.Text & ""
        'end perintah delete
    Else
        Exit Sub
    End If
End If

```

Gambar 2.12 Jendela Kode (Kurniawan, 2010)

g. Event

Menurut Juanita (2009) *Event* merupakan suatu kejadian yang akan diterima oleh suatu objek. *Event* yang diterima oleh objek berfungsi untuk menjalankan kode program yang ada di dalam objek tersebut.

h. Method

Menurut Juanita (2009) *Method* adalah suatu kumpulan perintah yang memiliki kegunaan yang sama dengan fungsi atau sebuah prosedur, tetapi perintah-perintah tersebut sudah disediakan dalam suatu objek.

i. Module

Menurut Juanita (2009) *Module* hampir sama fungsinya dengan *form*, tetapi *module* tidak berisi objek dan bentuk standar, dan *module* berisi kode program atau prosedur yang dapat digunakan oleh program aplikasi.

4. Struktur Program Visual Studio 2010

Menurut Juanita (2009) Secara umum struktur program *Visual Basic 2010* terdiri dari dua bagian yaitu bagian deklarasi program dan bagian pernyataan program.

a. Bagian Deklarasi

Menurut Juanita (2009) Bagian ini adalah bagian yang berfungsi untuk meletakkan semua deklarasi data yang akan digunakan.

Secara umum kata cadangan yang merupakan bagian dari deklarasi adalah : *Dim*, *Public*, *Private*, *Const* dan *Type*.

1) Deklarasi *Dim* atau *Dimension*

Adalah kata cadangan yang sering dipakai untuk mendeklarasikan *variabel* yang akan digunakan dalam *Visual Basic*.

2) *Public* merupakan pernyataan yang menggantikan pernyataan global dalam *Visual Basic*.

3) Deklarasi *Private*

Private menyatakan bahwa semua *variabel* yang dideklarasikan oleh pernyataan ini berlaku secara khusus (*private*). Pernyataan *private* merupakan pernyataan level sub-program.

4) Deklarasi *Const* atau *Constanta*

Deklarasi ini sering dipakai untuk memberikan harga konstanta pada suatu *variabel*. *Const* merupakan pernyataan level module, artinya pernyataan ini pada dasarnya sering dipadukan dengan pernyataan *public* dan *private*.

5) Deklarasi *Type*

Type digunakan pada bagian deklarasi untuk mendefenisikan tipe data terdefenisi. Dimana *type* data ini mengandung satu atau lebih dari suatu *type* data.

b. Bagian Pernyataan

Menurut Juanita (2009) Program pada bagian ini ditulis pada jendela kode, jendela modul maupun jendela sub-program. Program pada bagian ini dapat memuat semua pemrograman pengendali saat *event driven programming* yang merupakan salah satu kelebihan pada *Visual Basic*.

5. Tipe Data Pemrograman Visual Basic 2010

Menurut Juanita (2009) Sebelum memulai pemrograman dengan *Visual Basic*, maka ada baiknya terlebih dahulu mengenal tipe data yang dipakai dalam *Visual Basic*.

Tabel 2.7 Tipe-tipe Data Visual Basic 2010 (Juanita, 2009)

TIPE	SIMBOL	JANGKAUAN	UKURAN
Boolean	Tidak Ada	True atau False	2 Byte
Byte	Tidak Ada	0 s/d 255	1 Byte
Integer	%	-32.768 s/d 32.7672	2 Byte
Long Integer	&	-2.147.486.648 s/d 2.147.483.647	4 Byte
Singel	!	-3.402823E s/d -401298-E-45	4 Byte
Double	#	-1.79769313486232E308 s/d -4.940564584124E-324	8 Byte
Currency	@	-922.337.203.685.477.5808 s/d -922.337.203.685.477.5808	8 Byte
Date	Tidak Ada	1 Januari 100 s/d 31 Desember 9999	8 Byte
Object	Tidak ada	Semua objek	4 Byte
Text/String	\$	1 s/d 2 triliyun, untuk Win 3.1 hanya sampai 65400	Sepanjang teksnya + 10 byte
Variant	Tidak ada	Semua angka yang mungkin sebesar tipe Double	Sepanjang string dan teks
User Defined (ditentukan oleh pemakai)	Tidak Ada	Sepanjang yang telah didefinisikan oleh pemrograman	Sepanjang yang dibutuhkan oleh defenisi tersebut

BAB III

ANALISA DAN HASIL

A. Analisa Sistem Yang Sedang Berjalan

Analisa sistem yang sedang berjalan merupakan pedoman untuk merancang sistem yang baru, sebab dengan menganalisa sistem yang sedang berjalan dapat diketahui kelemahan-kelemahan dari sistem yang lama dan keunggulan dari sistem yang baru. Sistem yang lama akan dijadikan perbandingan terhadap sistem yang baru yang akan diterapkan. Analisa sistem bertujuan mencari pemecahan masalah yang dihadapi pada sistem lama tersebut agar masalah yang sama tidak terjadi lagi pada masa yang akan datang.

Dalam analisis dan perencanaan sistem ini, kegiatan yang akan dilakukan adalah melakukan penelitian terhadap sistem yang sedang berjalan untuk data nyata secara detail sesuai dengan fakta-fakta yang ada dalam penelitian. Perkembangan suatu sistem sering kali dipengaruhi oleh perubahan kondisi yang dihadapi.

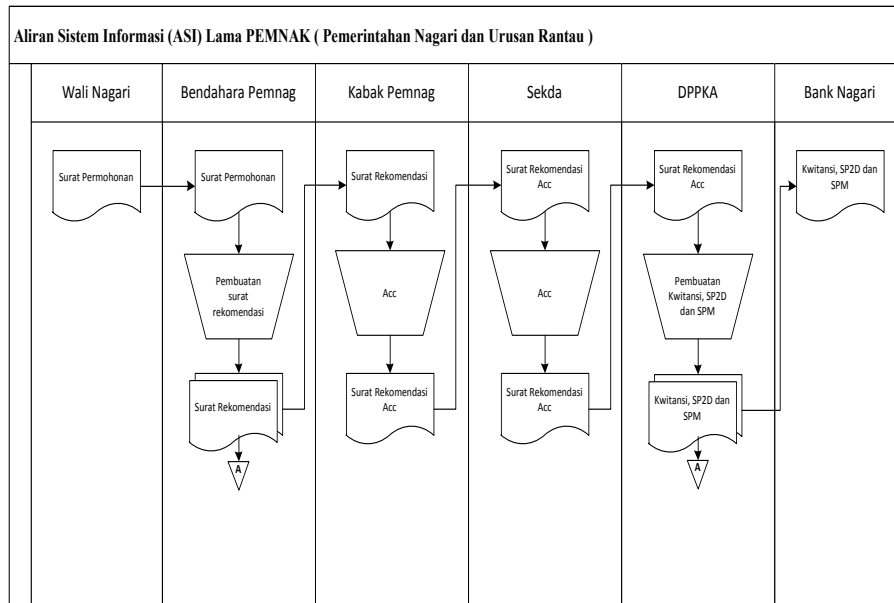
Salah satu faktor penyebabnya adalah penambahan jumlah data yang akan diolah untuk menghasilkan informasi yang dibutuhkan. Hal ini akan mengakibatkan sistem lama kewalahan dalam mengolah data dan akhirnya menyebabkan terjadinya ketidakakuratan data yang dihasilkan. Sistem Informasi saat ini masih dilakukan secara manual seperti yang dijabarkan pada BAB I.

1. Aliran Sistem Informasi

Aliran Sistem Informasi (ASI) merupakan aliran dari semua dokumen maupun laporan-laporan yang terjadi dalam proses pengolahan data laporan keuangan Pemerintah Nagari dan Urusan Rantau di Pemerintahan Kabupaten Tanah Datar. Aliran Sistem Informasi yang

sedang berjalan pada Pemerintahan Nagari dan Urusan Rantau di Pemerintahan Kabupaten Tanah Datar adalah:

- a. Nagari membuat surat permohonan ke Pemerintahan Nagari dan Urusan rantau
- b. Kemudian Pemerintahan Nagari dan Urusan Rantau membuat surat rekomendasi
- c. Setelah membuat surat rekomendasi
- d. Surat rekomendasi di ketahui Kabak Pemerintahan Nagari dan Urusan Rantau
- e. Setelah diketahui Kabak, dan ACC surat rekomendasi di berikan ke Sekda untuk di ketahui
- f. Surat rekomendasi yang telah diketahui dan ACC, surat disampaikan ke DPPKA (Dinas Pendapatan Pengelolaan Keuangan dan Aset)
- g. Setelah disampaikan ke DPPKA, DPPKA menyetujui dan membuat kwitansi, SP2D (Surat Perintah Pembayaran Dana), SPM (Surat Perintah Pembayaran)
- h. Kemudian kwitansi, SP2D, dan SPM diberikan ke Bank Nagari
- i. Bank Nagari menyetujui, dan dana di transfer ke rekening umum Nagari
- j. Stelah di transfer, Nagari berhak mengambil sesuai kebutuhan Nagari
- k. Dana yang di ambil, harus dipertanggungjawabkan sesuai kebutuhan Nagari
- l. Pada akhir tahun akan dilakukan evaluasi oleh badan-badan yang terkait.



Gambar 3.1 Aliran Sistem Informasi Yang Sedang Berjalan

2. Evaluasi Sistem Yang Sedang Berjalan

Evaluasi sistem ini dilakukan untuk mengetahui masalah yang akan terjadi pada sistem yang lama sebagai dasar untuk merancang sistem yang baru. Sistem yang ada sudah berjalan dengan cukup baik, namun dirasa kurang efektif dan efisien serta menghasilkan data yang kurang akurat.

Kelemahan sistem yang lama adalah:

- a. Laporan keuangan yang dilakukan dibuku besar atau secara manual akan menyulitkan pegawai.
- b. Bertumpuknya pengarsipan laporan keuangan yang akan merepotkan pegawai bila dilakukan peninjauan ulang data.

B. Desain Sistem Baru

Setelah pemecahan masalah dilakukan pada tahap analisa, maka perancangan sistem baru dilakukan. Tujuan dari perancangan sistem baru ini sebagai penyempurnaan dari sistem yang sedang berjalan saat ini. Adapun

hal-hal yang perlu dirancang dalam desain ini tidak terlepas dari bentuk yang telah ada sebelumnya. Untuk merancang sistem yang baru tentu berpedoman pada sistem yang lama. Pada perancangan ataupun desain sistem yang baru diharapkan adanya perbaikan dan penyempurnaan dari sistem yang lama secara keseluruhan.

1. Desain Global

Desain sistem secara umum atau disain global dapat didefinisikan sebagai suatu gambaran, perencanaan dan pembuatan sketsa pengaturan dari beberapa elemen yang terpisah dalam satu kesatuan yang utuh dan sesuai dengan fungsinya.

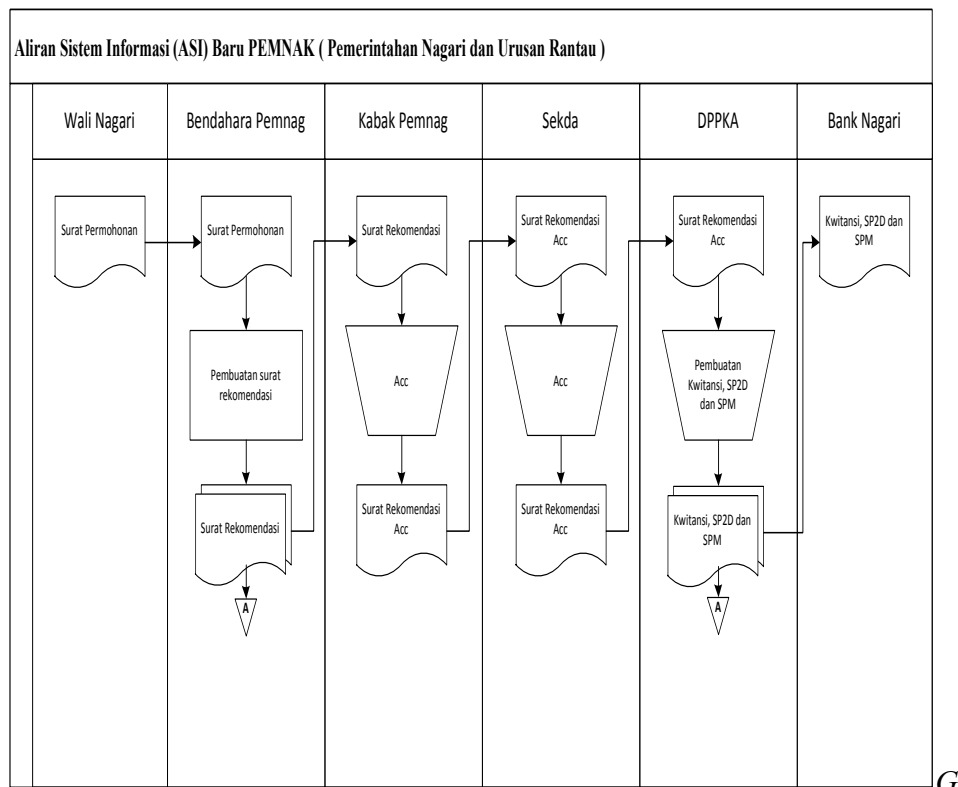
a. Aliran Sistem Informasi Baru

Pada dasarnya ASI baru tidak berbeda jauh dari ASI Lama. Hanya saja proses pengolahan laporan keuangan Pemerintahan Kabupaten Tanah Datar dilakukan dengan bantuan komputer yang ditunjang dengan program *Visual Basic Ultimate* yang dilengkapi dengan *database Access 2010*.

Aliran Sistem Informasi(ASI) yang diusulkan adalah:

- a. Nagari membuat surat permohonan ke Pemerintahan Nagari dan Urusan rantau
- b. Kemudian Pemerintahan Nagari dan Urusan Rantau membuat surat rekomendasi
- c. Setelah membuat surat rekomendasi
- d. Surat rekomendasi di ketahui Kabak Pemerintahan Nagari dan Urusan Rantau
- e. Setelah diketahui Kabak, dan ACC surat rekomendasi di berikan ke Sekda untuk di ketahui
- f. Surat rekomendasi yang telah diketahui dan ACC, surat disampaikan ke DPPKA (Dinas Pendapatan Pengelolaan Keuangan dan Aset)

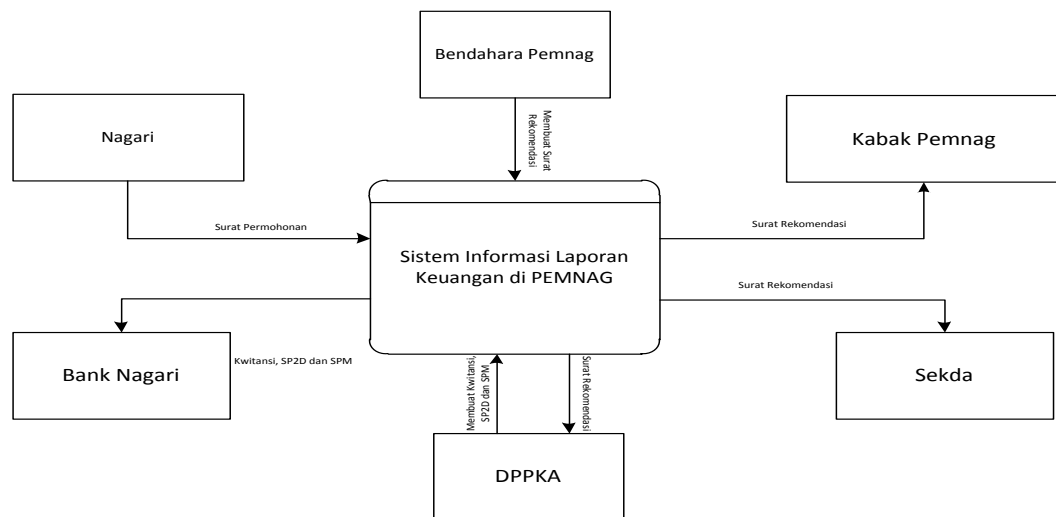
- g. Setelah disampaikan ke DPPKA, DPPKA menyetujui dan membuat kwitansi, SP2D (Surat Perintah Pembayaran Dana), SPM (Surat Perintah Pembayaran)
- h. Kemudian kwitansi, SP2D, dan SPM diberikan ke Bank Nagari
- i. Bank Nagari menyetujui, dan dana di transfer ke rekening umum Nagari
- j. Stelah di transfer, Nagari berhak mengambil sesuai kebutuhan Nagari
- k. Dana yang di ambil, harus dipertanggungjawabkan sesuai kebutuhan Nagari
- l. Pada akhir tahun akan dilakukan evaluasi oleh badan-badan yang terkait.



ambar 3.2 Aliran Sistem Informasi (ASI) Baru

b. Context Diagram

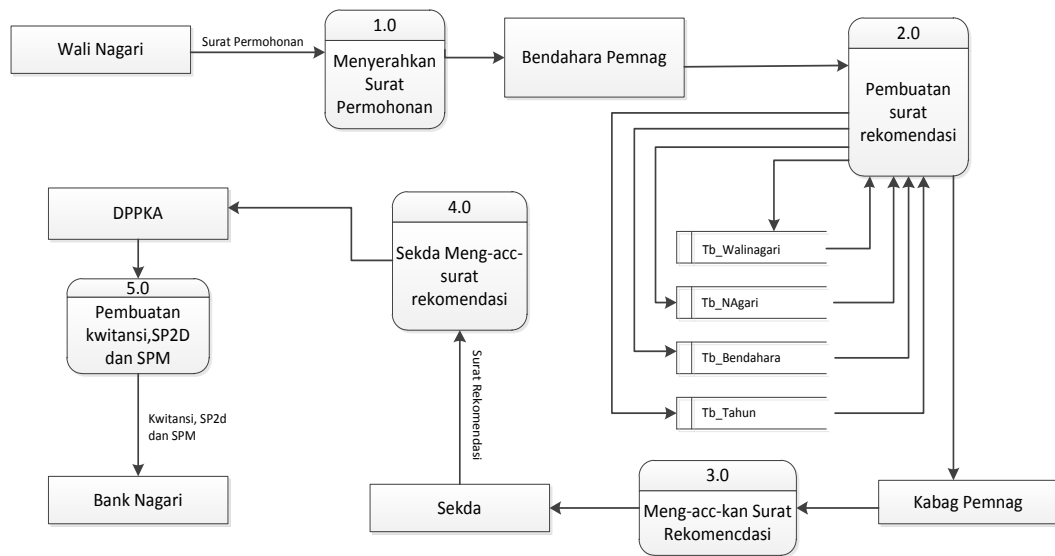
Context Diagram merupakan alat bantu perancangan secara global yang memperlihatkan sistem secara umum dan bagian-bagian dari sub sistem-sub sistem yang terlibat di dalam sistem secara keseluruhan, keterkaitan dan interaksi antar sub sistem-sub sistem. Untuk lebih jelasnya dapat dilihat pada gambar 3.3 berikut ini:



Gambar 3.3 *Context Diagram*

c. Data Flow Diagram

Data Flow Diagram (DFD) adalah sebuah alat dokumentasi grafik yang menggunakan nomor kecil dari simbol untuk menggambarkan bagaimana aliran data, mengakhiri hubungan dalam suatu proses. DFD dapat dilihat pada gambar 3.4 sebagai berikut:



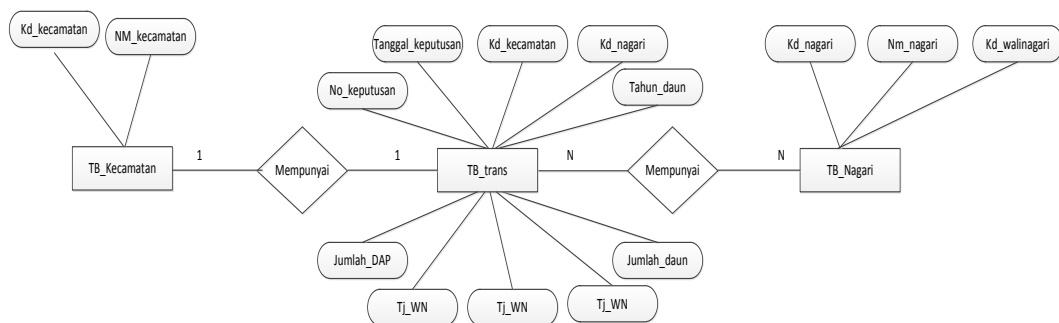
Gambar 3.4 Data Flow Diagram

Dilihat dari Aliran Sistem Informasi yang baru, maka terdapat 6 entity, dan pada Data Flow Diagram di atas juga terdiri dari 6 entity yaitu Wali Nagari, Bendahara, Kabak, Sekda, DPPKA dan Bank Nagari..

Kegiatan pengelolaan laporan keuangan di mulai dari Nagari membuat surat permohonan ke Pemerintahan Nagari dan Urusan Rantau. Pemerintahan Nagari dan Urusan Rantau membuat surat rekomendasi yang diketahui Kabak, dan Sekda. Setelah diketahui dan di ACC surat rekomendasi di sampaikan ke DPPKA (Dinas Pendapatan Pengelolaan Keuangan dan Aset). DPPKA menyetujui dan membuat kwitansi, SP2D (Surat Perintah Pembayaran Dana), SPM (Surat Perintah Pembayaran). Kemudian kwitansi, SP2D, dan SPM diberikan ke Bank Nagari. Bank Nagari menyetujui, dan dana di transfer ke rekening umum Nagari Setelah di transfer, Nagari berhak mengambil sesuai kebutuhan Nagari. Dana yang di ambil, harus dipertanggungjawabkan sesuai kebutuhan Nagari. Pada akhir tahun akan dilakukan evaluasi oleh badan-badan yang terkait.

d. Entity Relationship Diagram (ERD)

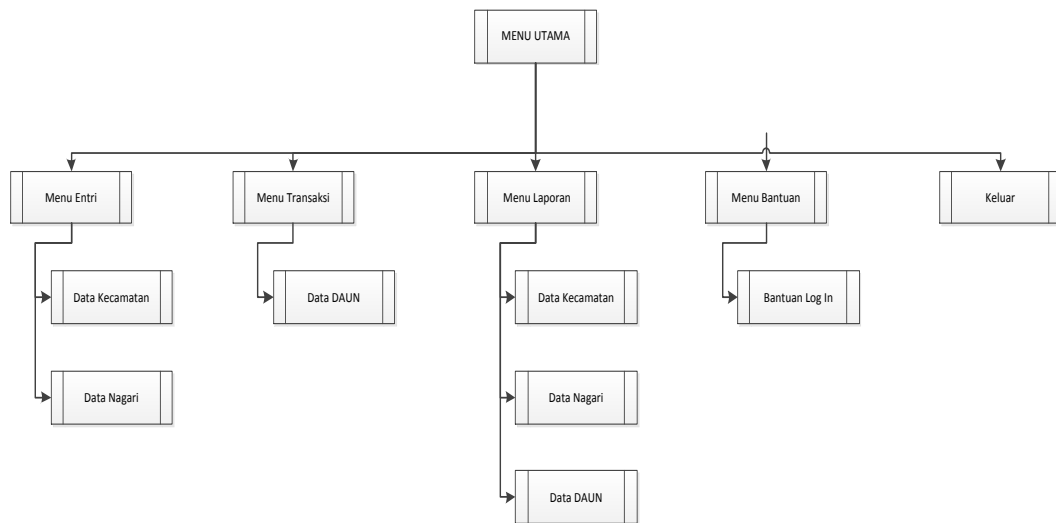
Entity Relationship Diagram (ERD) menggambarkan tentang hubungan antara suatu file dengan file yang lainnya, dengan menggunakan suatu penghubung yaitu relasi. Adapun bentuk dari Entity relationship diagram yang dirancang dapat lihat gambar 3.5 berikut :



Gambar 3.5 Entity Relationship Diagram

e. Struktur Program

Setelah menganalisa sistem yang sedang berjalan serta melakukan penelitian, maka dapat dirancang suatu sistem informasi baru yang diharapkan dapat meningkatkan efektifitas dan efisien kerja pada bagian tersebut, dimana keseluruhan dari sistem tersebut tertuang dalam bentuk program aplikasi. Adapun struktur program yang dirancang adalah sebagai berikut :



Gambar 3.6 Struktur Program

2. Desain Terinci

a. Desain Output

Desain *Output* merupakan bentuk laporan yang dihasilkan sistem yang dirancang sedemikian rupa sehingga dapat digunakan untuk kemajuan suatu usaha dan dapat dipakai sebagai bahan perbandingan oleh pimpinan dalam mengambil keputusan. Adapun desain *output* yang telah penulis rancang adalah sebagai berikut :

Adapun desain output yang telah penulis rancang adalah sebagai berikut :

1. Laporan Data Kecamatan

Laporan data Kecamatan berguna untuk memudahkan bendahara Pemerintahan Nagari mengenai laporan data Kecamatan. Desain output laporan data Kecamatan dapat dilihat pada tabel 3.1 berikut:

Tabel 3.1 Laporan Data Kecamatan

<div style="border: 1px solid black; width: 100px; margin: 0 auto; padding: 5px;">Logo</div> <div style="border: 1px solid black; width: 250px; margin: 10px auto; padding: 5px;">Nama Instansi</div>	
Laporan Data Kecamatan	
Kode Kecamatan	Nama Kecamatan
<div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(30) </div> <div style="text-align: center; height: 100px; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; border-bottom: 1px solid black;"></div> </div> <div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(30) </div>	<div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(50) </div> <div style="text-align: center; height: 100px; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; border-bottom: 1px solid black;"></div> </div> <div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(50) </div>

Batusangkar, DD-MM-YYYY
Tertanda
Bupati Tanah Datar

2. Laporan Data Nagari

Laporan data Nagari berfungsi untuk memudahkan bendahara Pemerintahan Nagari untuk mengetahui jumlah Nagari. Desain output laporan data Nagari dapat dilihat pada tabel 3.2 berikut:

Tabel 3.2 Laporan Data Nagari

<div style="border: 1px solid black; width: 100px; margin: 0 auto; padding: 5px;">Logo</div> <div style="border: 1px solid black; width: 250px; margin: 10px auto; padding: 5px;">Nama Instansi</div>	
Laporan Data Nagari	
Kode Nagari	Nama Nagari
<div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(30) </div> <div style="text-align: center; height: 100px; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; border-bottom: 1px solid black;"></div> </div> <div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(30) </div>	<div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(50) </div> <div style="text-align: center; height: 100px; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; border-bottom: 1px solid black;"></div> </div> <div style="text-align: center;"> <hr style="width: 80%; margin: 0 auto;"/> X(50) </div>

Batusangkar, DD-MM-YYYY
Tertanda
Bupati Tanah Datar

3. Laporan Data Transaksi DAUN

Laporan data transaksi berguna untuk menyampaikan informasi mengenai data dan keterangan bendahara yang memberikan laporan data transaksi. Desain output laporan data transaksi DAUN dapat dilihat pada tabel 3.3 berikut:

Tabel 3.3 Laporan Data Transaksi DAUN

<div style="border: 1px solid black; width: 60px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;">Logo</div> <div style="border: 1px solid black; width: 280px; height: 25px; margin: 10px auto; text-align: center;">Nama Instansi</div>									
Laporan Data Transaksi DAUN									
No Keputusan	Tanggal Keputusan	Tahun DAUN	Kode Kecamatan	Kode Nagari	Jumlah DAP	T. Wali Nagaari	T. BPRN	RASKIN	Jumlah DAUN
X(30)	DD-MM-YYYY	X(4)	X(30)	X(30)	X(15)	X(15)	X(15)	X(15)	X(15)
/	/	/	/	/	/	/	/	/	/
X(30)	DD-MM-YYYY	X(4)	X(30)	X(30)	X(15)	X(15)	X(15)	X(15)	X(15)

Batusangkar, DD-MM-YYYY
Tertanda
Bupati Tanah Datar

b. Desain Input

Dalam setiap pemrosesan perlu ada data masukan, dimana data yang akan diproses harus dimasukkan terlebih dahulu, tentunya melalui *interface* (perangkat penghubung) antara pengguna dengan *hardware* dan *software*. Untuk itu agar memudahkan dan tidak terjadinya kesalahan pemasukan data, maka dirancang bentuk *menu* tampilan yang mudah digunakan

untuk memasukkan data tersebut. Berikut ini adalah bentuk rancangan yang telah dibuat :

1. Entri Data Kecamatan

Desain input data Kecamatan berguna untuk mempermudah dalam penginputan dan pencarian data Kecamatan. Desain input data Kecamatan dapat dilihat pada gambar 3.7 berikut:

Entri Data Kecamatan	
Logo	Nama Instansi
Kode Kecamatan	x(30)
Nama Kecamatan	x(50)
Simpan	Hapus
Batal	Keluar
Kode Kecamatan	Nama Kecamatan
x(30)	x(50)
x(30)	x(50)

Gambar 3.7 : Desain Form Entri Data Kecamatan

2. Entri Data Nagari

Desain input data Nagari berguna untuk mempermudah dalam penginputan dan pencarian data Nagari. Desain input data Nagari dapat dilihat pada gambar 3.8 berikut:

Entri Data Nagari	
Logo	Nama Instansi
Kode Nagari	X(30)
Nama Nagari	X(50)
Simpan	Hapus
Batal	Keluar
Kode Nagari	Nama Nagari
X(30)	X(50)
X(30)	X(50)

Gambar 3.8 : Desain Form Entri Data Nagari

3. Entri Data Transaksi DAUN

Desain input data transaksi DAUN berguna untuk mempermudah dalam penginputan dan pencarian data transaksi DAUN. Desain input data transaksi DAUN dapat dilihat pada gambar 3.9 berikut:

Entri Data Transaksi DAUN

Logo Nama Instansi Logo

No Keputusan X(30)

Tanggal Keputusan DD-MM-YYYY Tahun DAUN X(4)

Kode Nagari X(30) X(50)

Kode Kecamatan X(30) X(50)

Jumlah DAP X(15) Rp. 0 T. Wali Nagari X(15) Rp. 0

T. BPRN X(15) Rp. 0 RASKIN X(15) Rp. 0

Jumlah DAUN X(15) Rp. 0

Simpan Hapus Batal Keluar

No Keputusan	Tanggal Keputusan	Tahun DAUN	Kode Kecamatan	Kode Nagari	Jumlah DAP	T. Wali Nagari	T. BPRN	RASKIN	Jumlah DAUN
X(30)	DD-MM-YYYY	X(4)	X(30)	X(30)	X(15)	X(15)	X(15)	X(15)	X(15)
X(30)	DD-MM-YYYY	X(4)	X(30)	X(30)	X(15)	X(15)	X(15)	X(15)	X(15)

Gambar 3.9 : Desain Form Entri Data Transaksi DAUN

c. Desain File

File merupakan kumpulan data-data atau *record-record* yang dibentuk oleh beberapa *field*. Desain *file* yang dirancang adalah:

1. File Data Transaksi DAUN

Tabel 3.4 Uraian *File* Transaksi DAUN

Database : dbDAUN

Table Name : tbTrans

File Key : no_keputusan, kd_Kec, kd_Nag

Fungsi : menyimpan data transaksi DAUN

Field Name	Type	Width	Description
no_keputusan	Text	30	Nomor Keputusan
tanggal_keputusan	Date	dd/mm/yyyy	Tanggal Keputusan
tahun_DAUN	Number	4	Tahun Dana Alokasi Umum Nagari
kd_Kec	Text	30	Kode Kecamatan

kd_Nag	Text	30	Kode Nagari
Jumlah_DAP	Number	15	Jumlah Dana Alokasi Pembangunan
Tj_WN	Number	15	Tunjangan Wali Nagari
Tj_BPRN	Number	15	Tunjangan Badan Perwakilan Rakyat Nagari
Jumlah_RASKIN	Number	15	Jumlah Beras Miskin
Jumlah_DAUN	Number	15	Jumlah Dana Alokasi Umum Nagari

2. File Data Kecamatan

Tabel 3.5 Uraian *File* Kecamatan

Database : dbDAUN
 Table Name : tbKecamatan
 File Key : kd_Kec
 Fungsi : menyimpan data Kecamatan

Field Name	Type	Width	Description
kd_Kec	Text	30	Kode Kecamatan
Kecamatan	Text	50	Kecamatan

3. File Data Nagari

Tabel 3.6 Uraian *File* Nagari

Database : dbDAUN
 Table Name : tbNagari
 File Key : kd_Nagari,
 Fungsi : menyimpan data Nagari

Field Name	Type	Width	Description
kd_Nagari	Text	30	Kode Nagari
Nagari	Text	50	Nagari

BAB IV

PENUTUP

Bab ini merupakan bab yang terakhir dari penulisan tugas akhir ini, yang mana pada bab ini berisikan kesimpulan dan saran-saran untuk dilakukan perbaikan-perbaikan yang dianggap perlu pada sistem yang ada pada saat ini. Penulis menyadari bahwa sistem yang diusulkan ini masih ada kelemahan-kelemahan dan kekurangan, namun sistem ini masih mempunyai kelebihan dan keunggulan dibandingkan dengan sistem yang sedang berjalan sekarang ini terutama dalam hal pengolahan data DAUN pada Pemerintahan Nagari dan Urusan Rantau di Pemerintahan Kabupaten Tanah Datar.

Dari penjelasan dan pembahasan pada bab-bab sebelumnya, maka penulis mencoba mengambil beberapa kesimpulan dan saran-saran dari uraian tersebut.

A. Kesimpulan

Dari hasil penelitian yang telah dilakukan dapat diambil suatu kesimpulan sebagai berikut :

1. Dengan rancangan sistem yang baru, yang menggunakan program aplikasi *Visual Studio 2010* dalam pengolahan data dapat membantu dalam meminimalkan kesalahan pengolahan data.
2. Dengan penerapan sistem baru ini semua data-data DAUN akan tersimpan dalam sebuah file database sehingga akan memudahkan dalam proses pencarian data.
3. Dengan menggunakan aplikasi program *Visual Studio 2010* dalam pengolahan data DAUN dapat menghasilkan laporan dengan cepat serta keakuratan yang lebih tinggi sehingga dapat membantu proses pengambilan data untuk masa yang akan datang.

B. Saran-saran

Dari hasil penelitian dan terdapatnya beberapa kelemahan yang ada pada sistem yang sedang berjalan, maka dapat dikemukakan beberapa saran yaitu :

1. Agar sistem yang dirancang dapat bekerja secara efektif dan efisien maka diperlukan tenaga terampil dalam pengoperasian aplikasi yang dibuat.
2. Untuk menghasilkan tenaga yang terampil perlu diadakan pelatihan terhadap pengguna sistem tentang bagaimana cara penggunaan sistem yang telah dirancang.
3. Dalam penerapan sistem komputerisasi sebaiknya didukung oleh perangkat atau alat yang memadai, baik dari segi manusia (*Brainware*) maupun segi peralatannya (*Hardware dan Software*).
4. Mencoba menggunakan sistem yang telah dirancang dan membandingkan dengan sistem yang lama, apabila ternyata sistem yang baru ini dianggap lebih efisien dan efektif maka disarankan pada Pemerintahan Nagari dan Urusan Rantau untuk mamakai sistem yang dirancang ini.
5. Dalam rancangan sistem yang baru ini diharapkan Pemerintahan Nagari dan Urusan Rantau mengevaluasi kembali data-data yang berhubungan dengan sistem baru ini. Jika ditemukan kekurangan-kekurangan maka sistem yang baru dirancang ini dapat diperbaiki kembali supaya lebih sempurna.

DAFTAR PUSTAKA

Tanah Datar (2008) *Tanah Datar Dalam Angka*, BPS (Badan Pusat Statistik Kabupaten Tanah Datar)

Al-Bahra, *Konsep Sistem Basis Data dan Implementasi*, Graha Ilmu, Yogyakarta: 2004

Faisal, M (2008). *Sistem Informasi Manajemen Jaringan*. Malang: UIN-Malang Press.

HM Jogyanto (2005). *Analisa dan Desain*. Yogyakarta: Andi Offset.

Jogyanto, HM, *Analisa dan Desain*, Andi, Yogyakarta : 2005

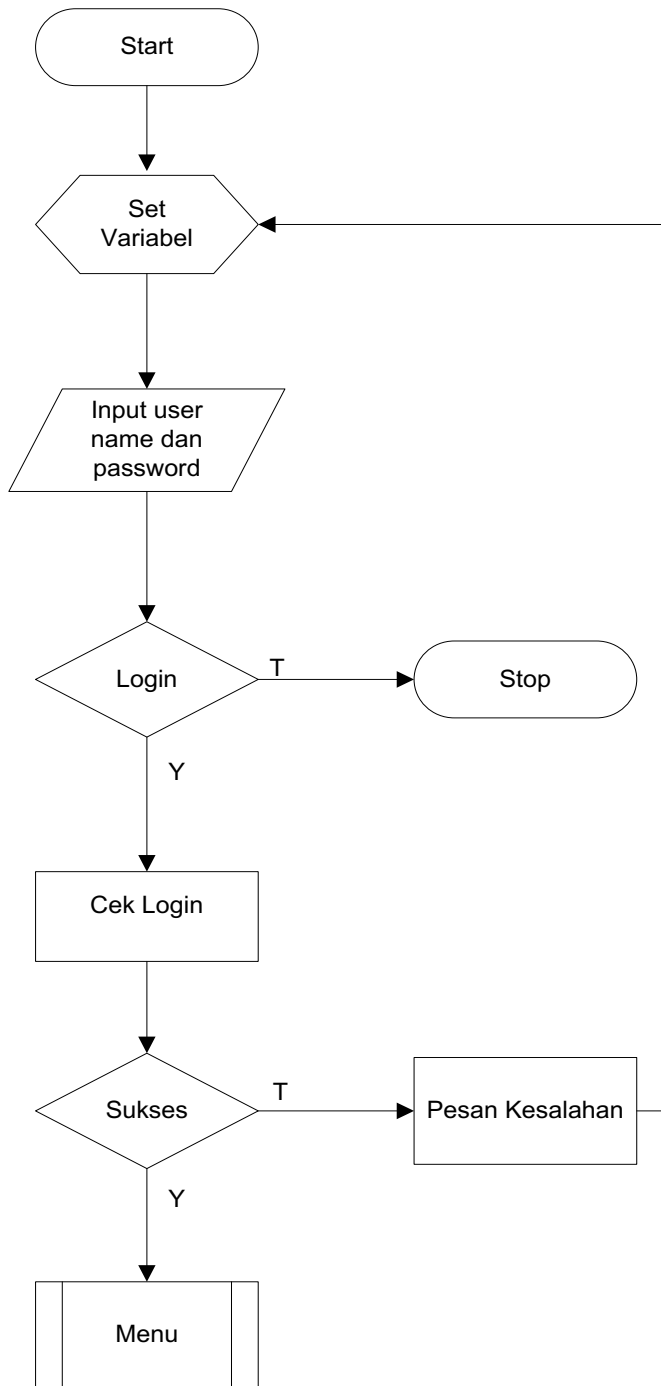
Kurniawan, Erick (2010). *Cepat Mahir Visual Basic 2010*. Yogyakarta: Andi Offset.

Ladjamuddin,(2006).*Rekayasa Perangkat Lunak*, Yogyakarta ,GRAHA ILMU

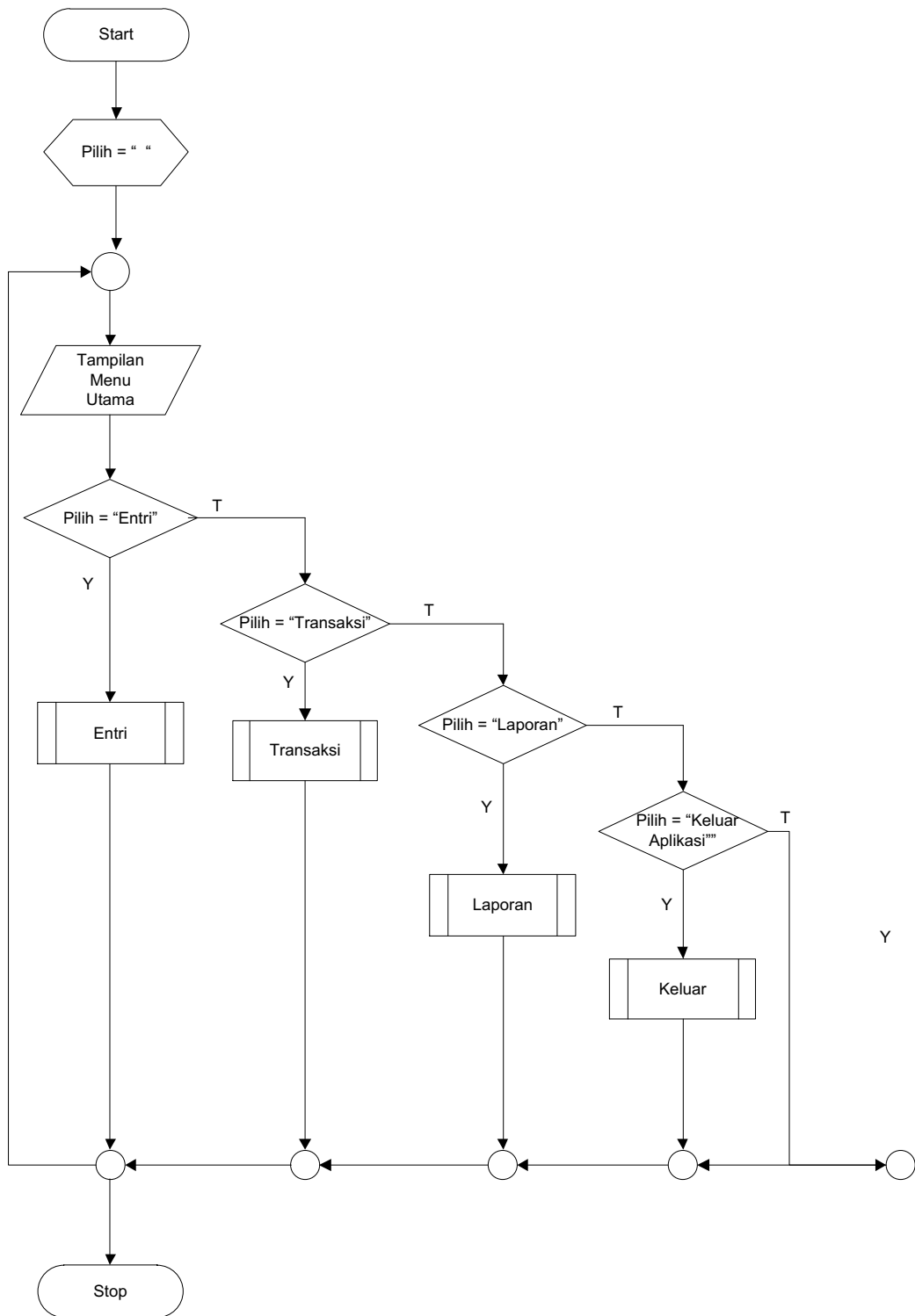
Sutabri, Tata, *Analisa Sistem Informasi*, Andi, Yogyakarta : 2004

Wahyono, Teguh, *Sistem Informasi (Konsep Dasar, Analisa Desain dan Implementasi)*, GRAHA ILMU, Yogyakarta : 2004

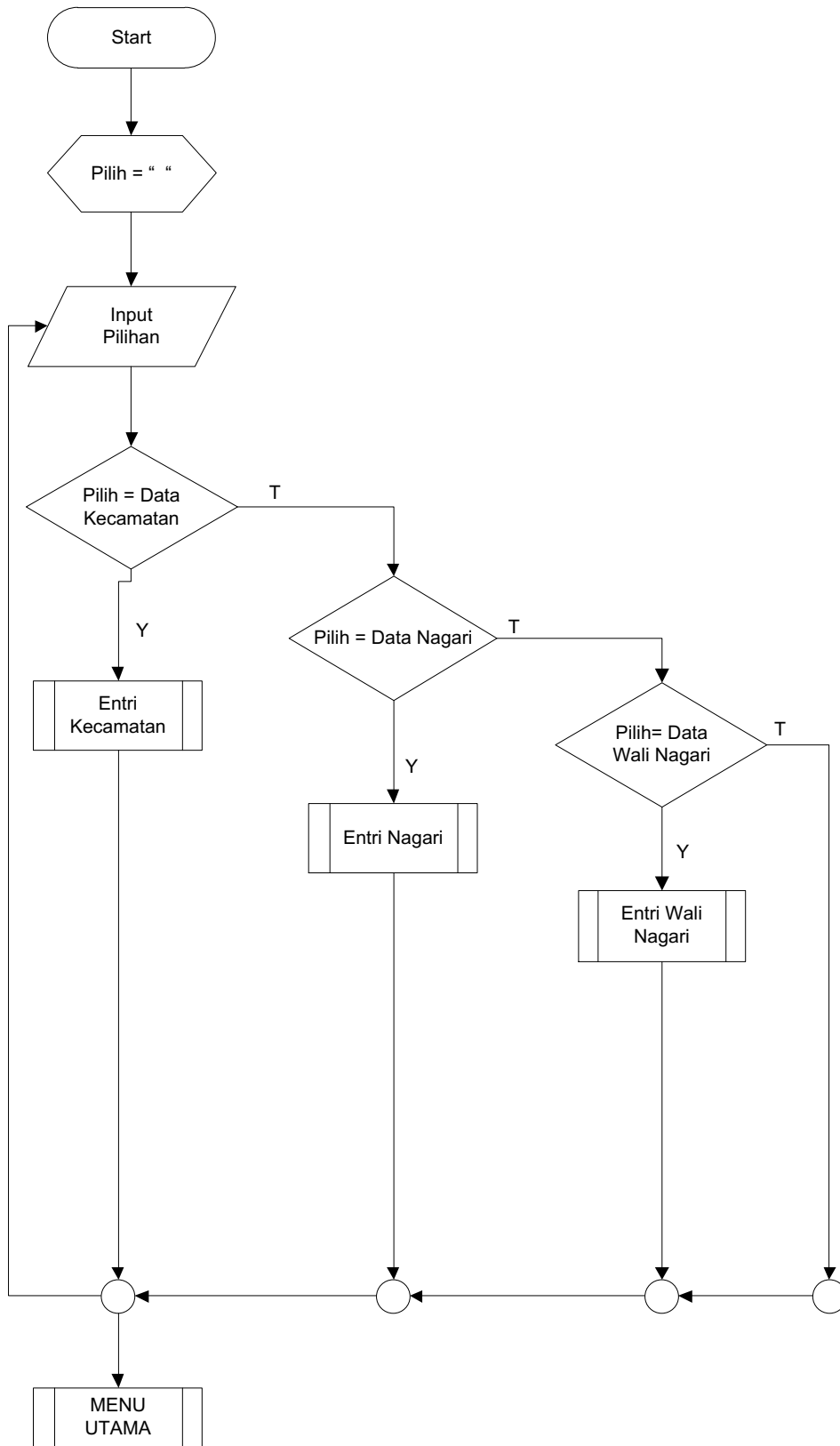
FLOWCHART LOGIN



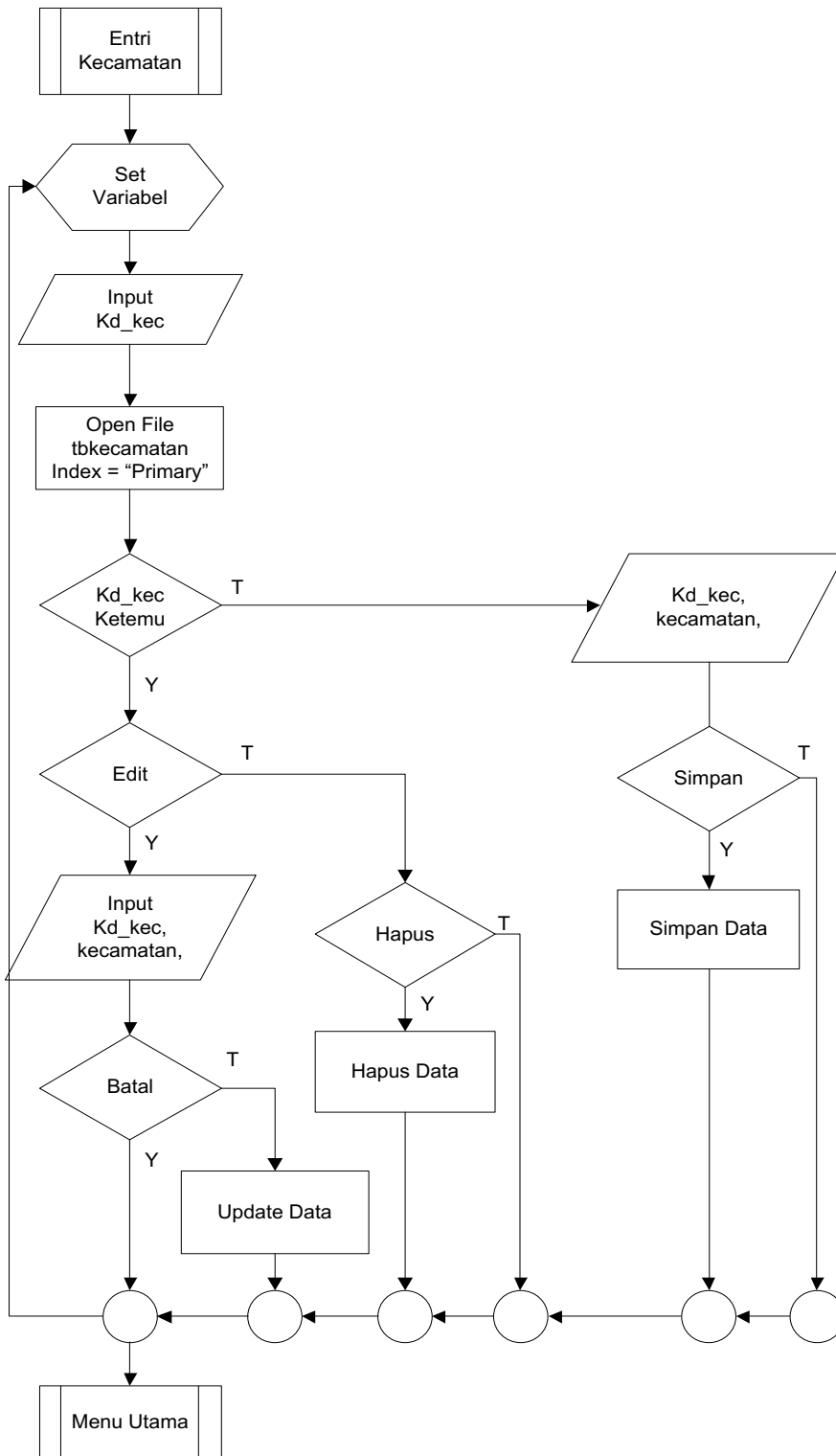
FLOWCHART MENU UTAMA



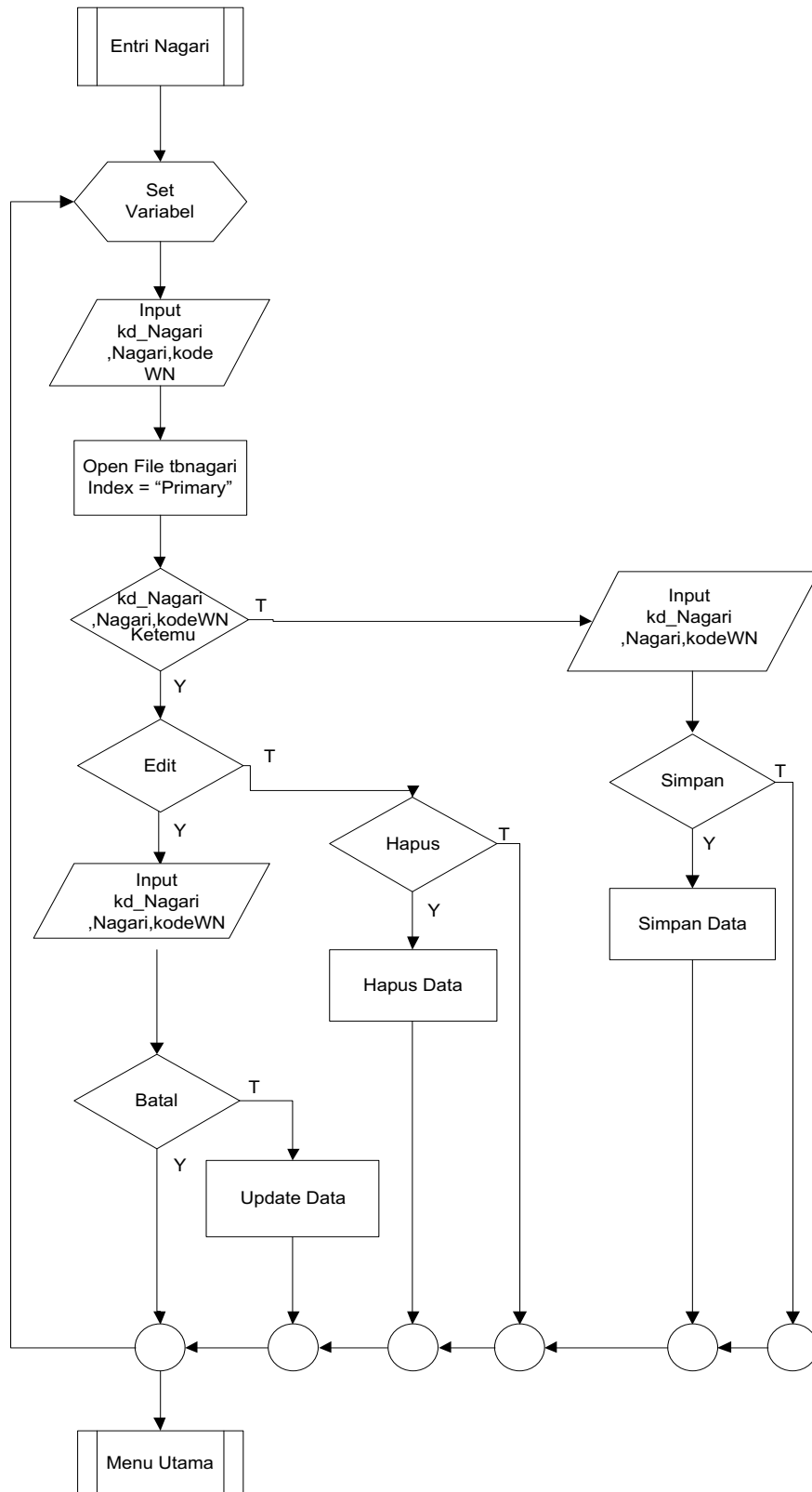
FLOWCHART MENU ENTRI



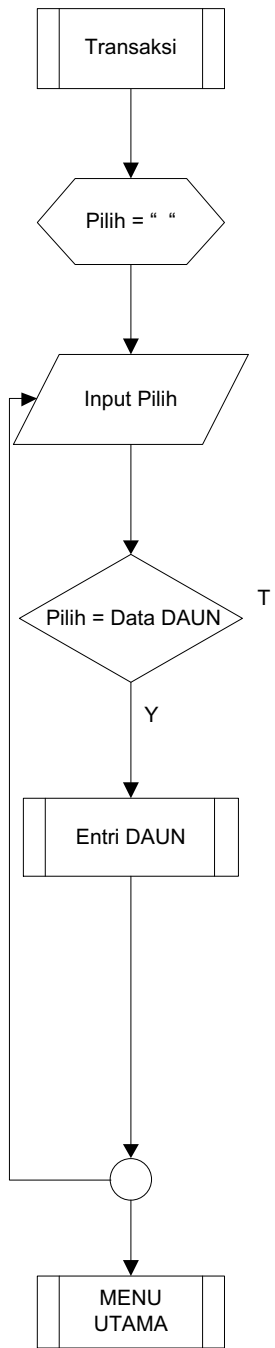
FLOWCHART ENTRI DATA KECAMATAN



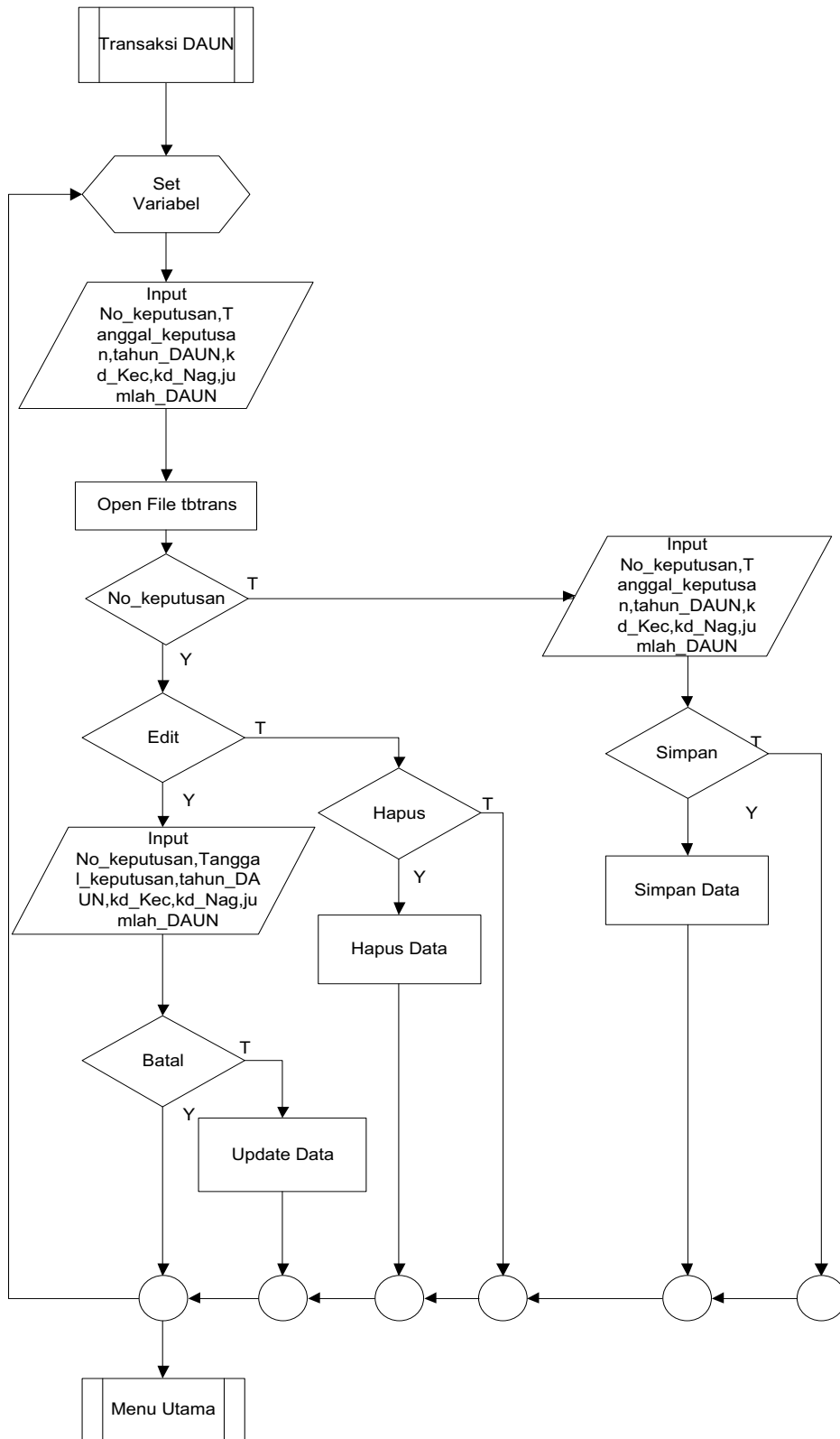
FLOWCHART ENTRI DATA NAGARI



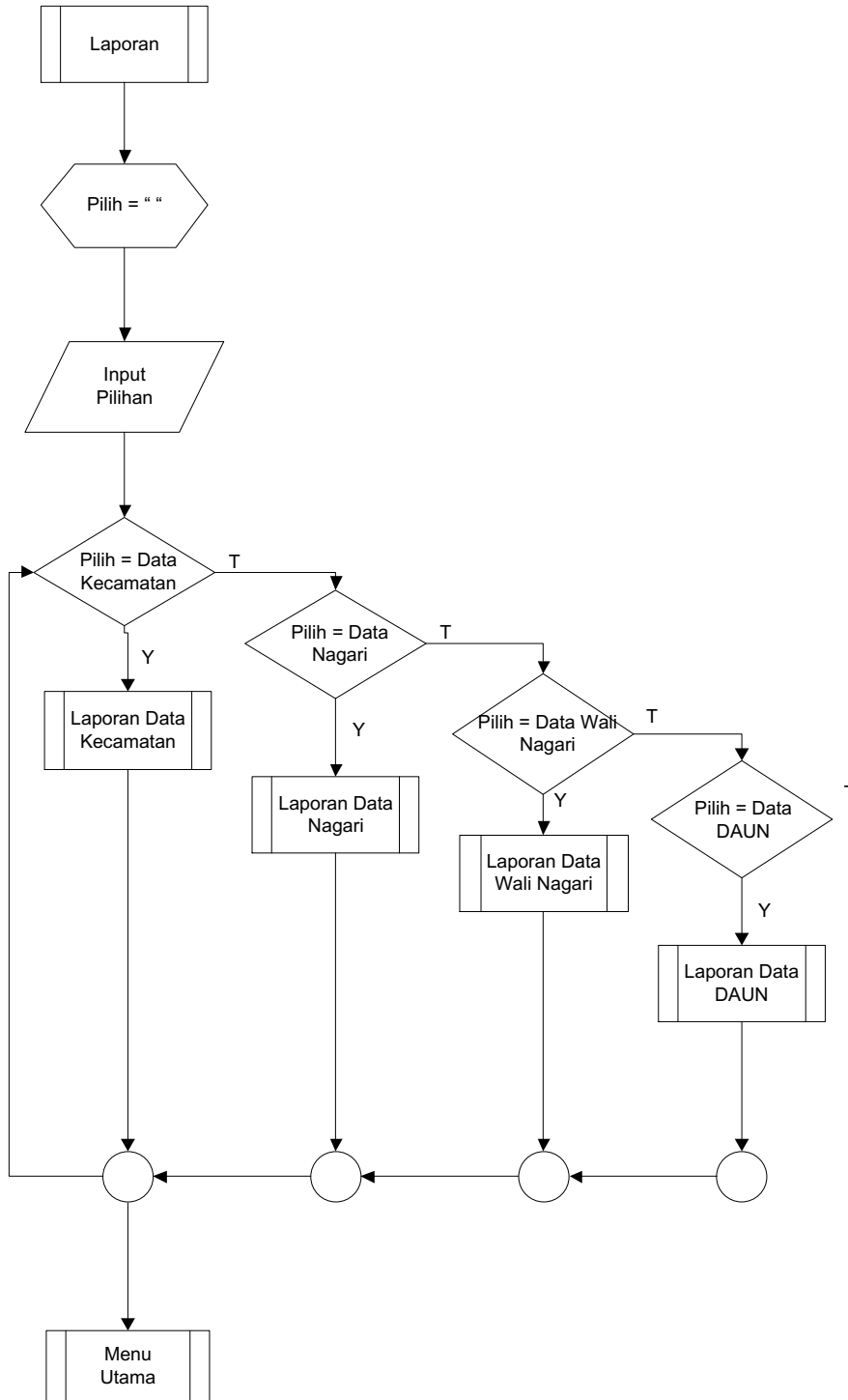
FLOWCHART MENU TRANSAKSI



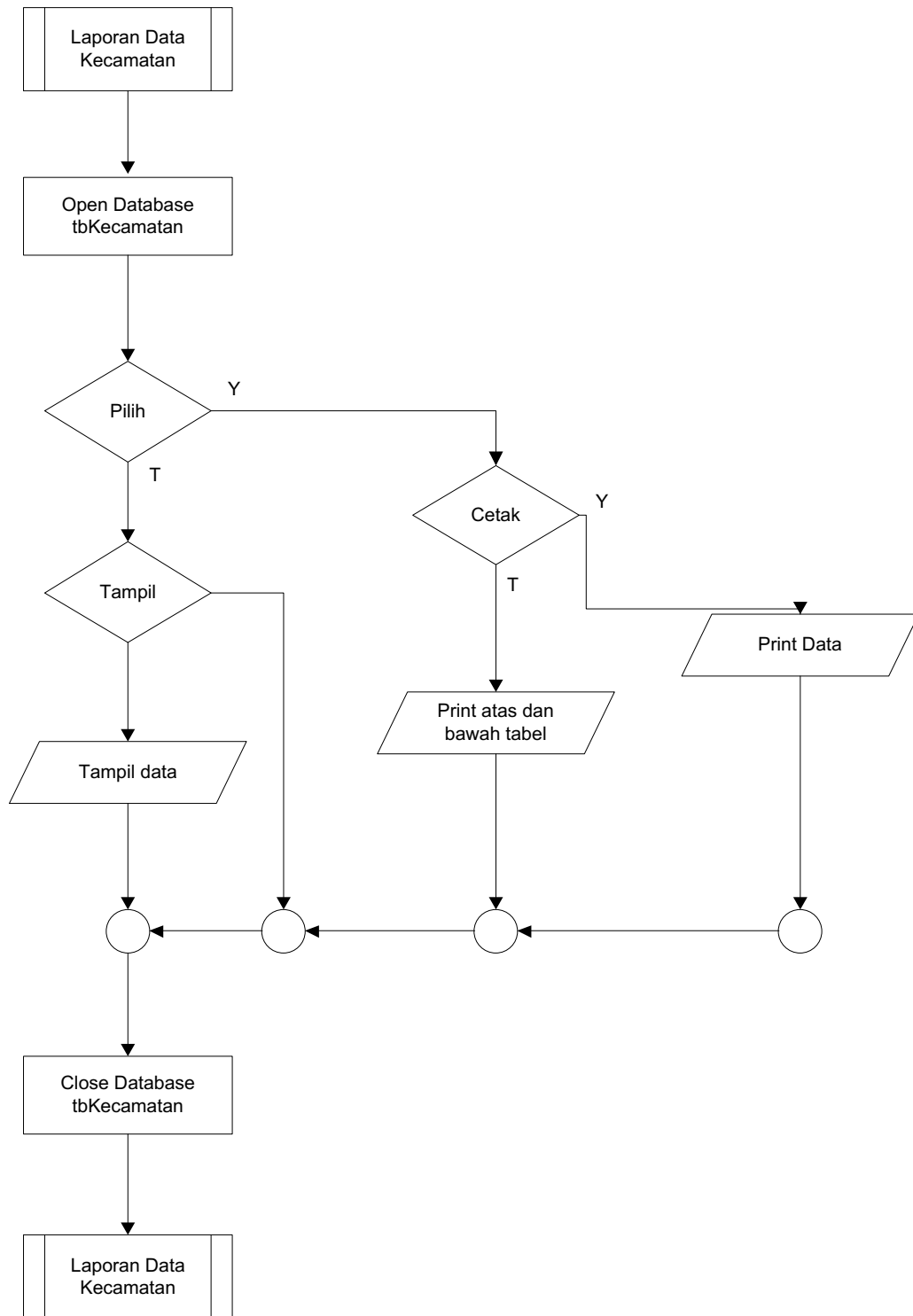
FLOWCHART ENTRI TRANSAKSI DAUN



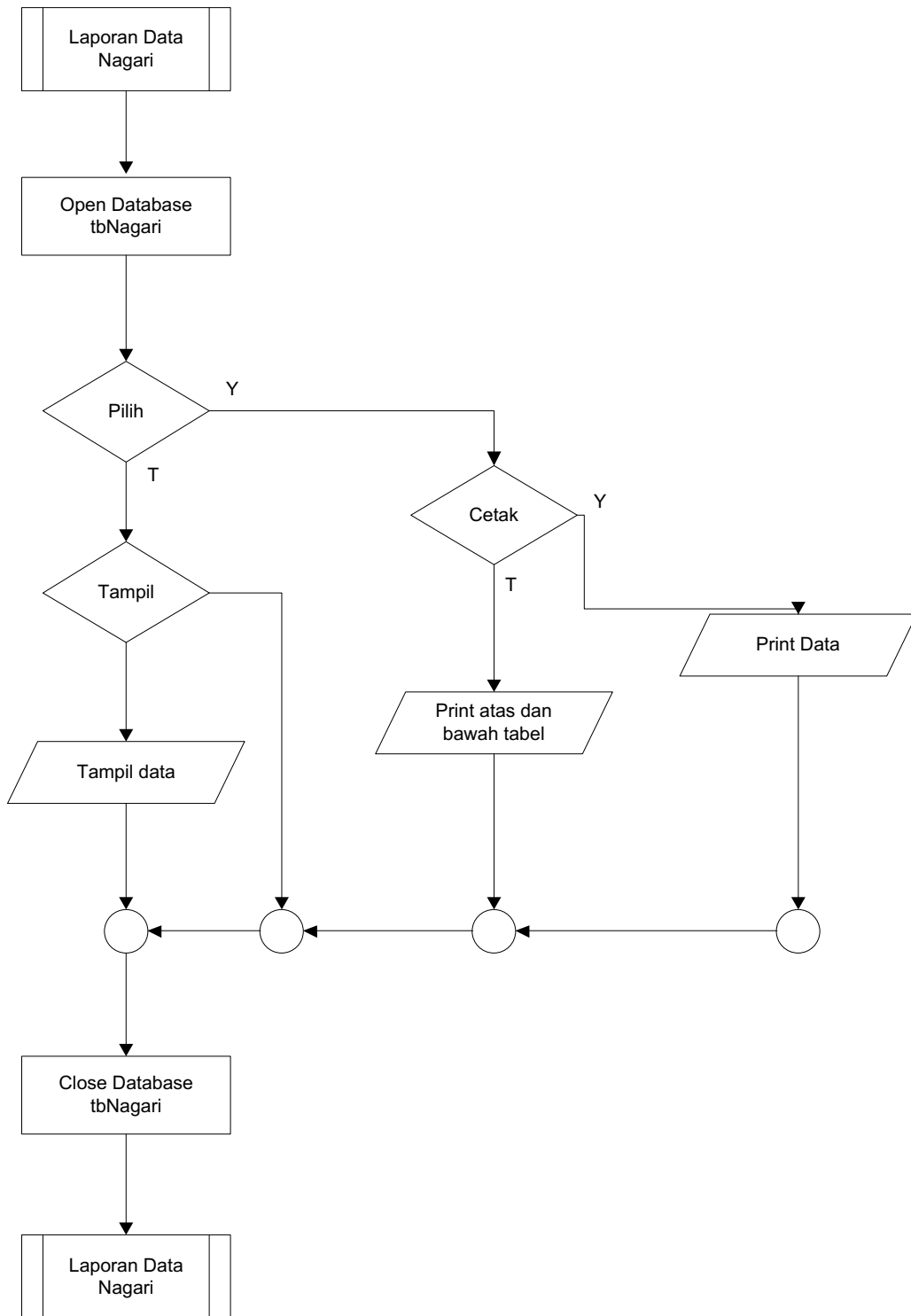
FLOWCHART MENU LAPORAN



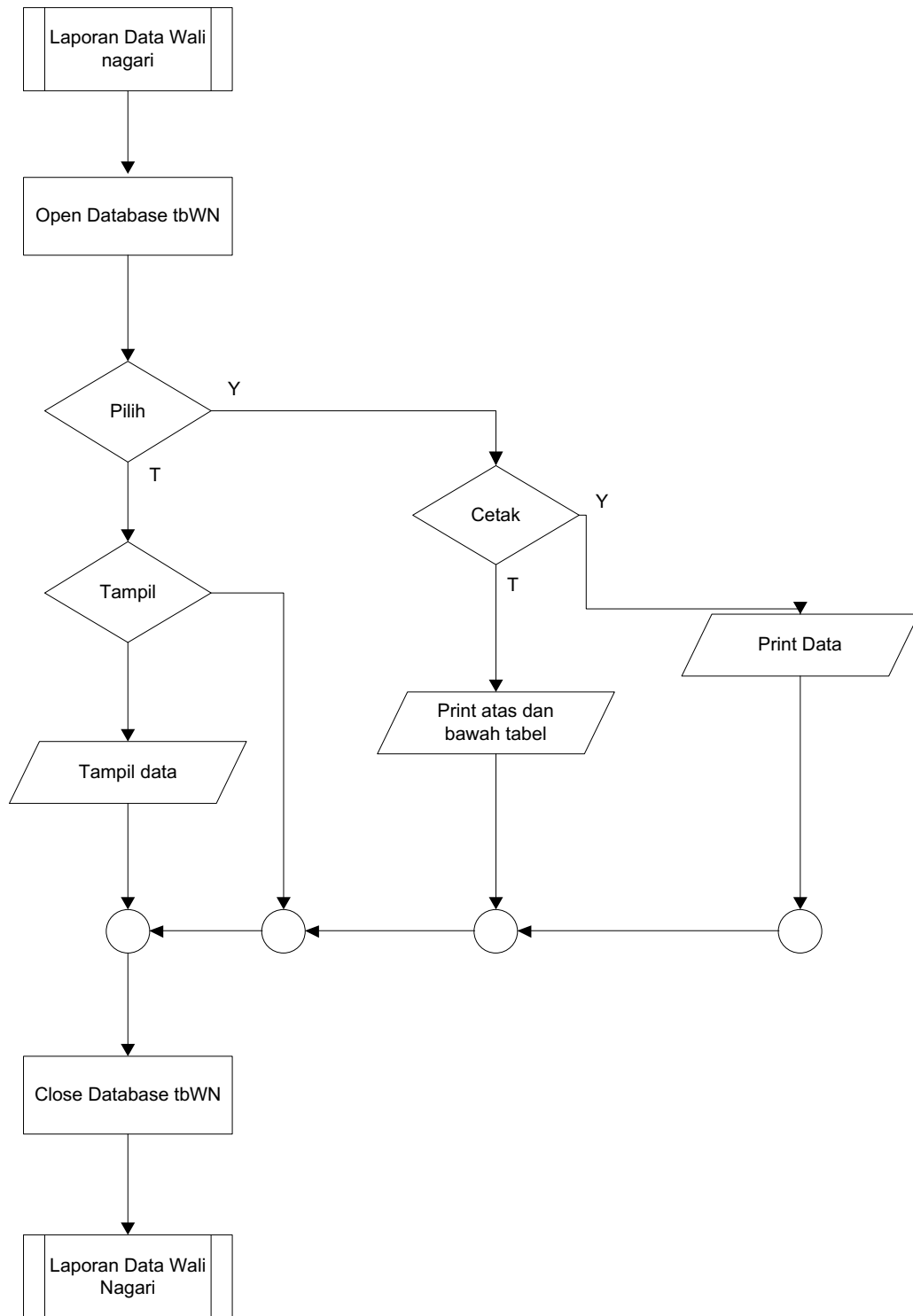
FLOWCHART LAPORAN DATA KECAMATAN



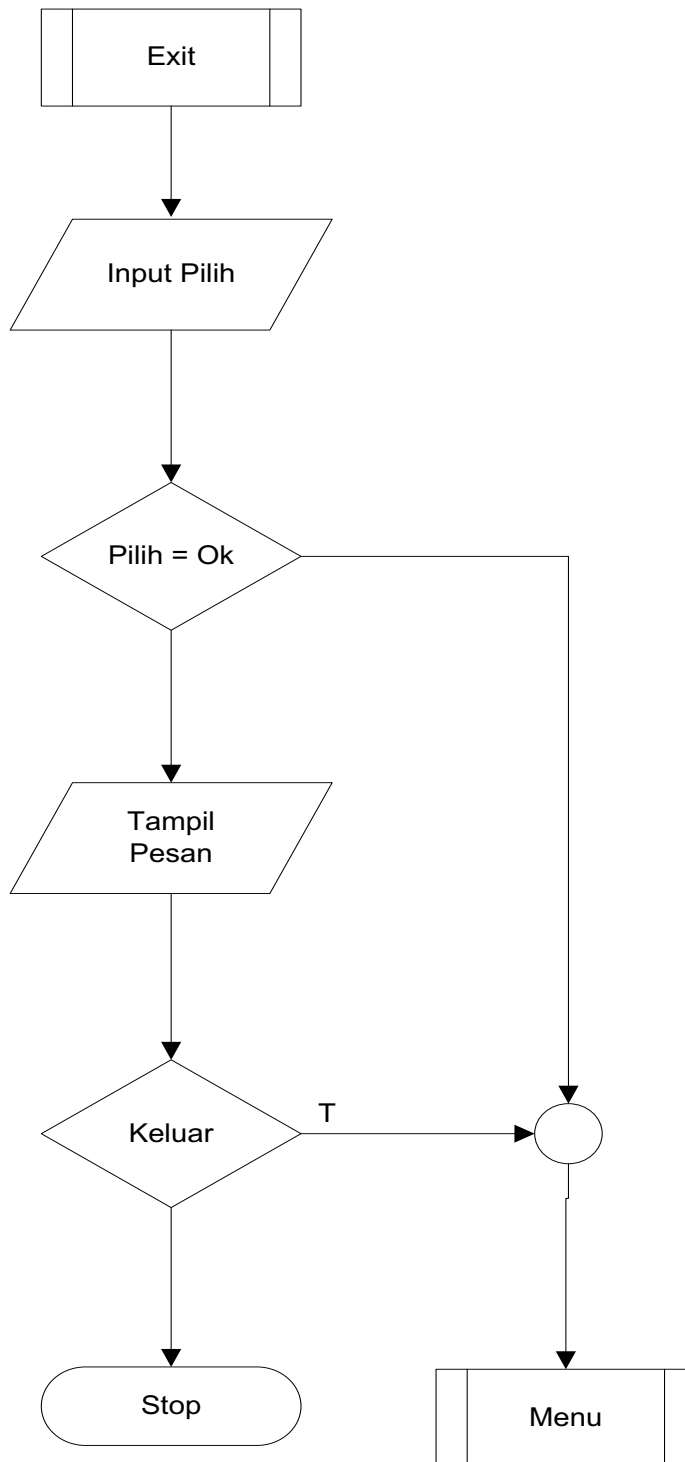
FLOWCHART LAPORAN DATA NAGARI



FLOWCHART LAPORAN DATA WALI NAGARI



MENU KELUAR



LAMPIRAN LISTING PROGRAM

1. mdlPro

```
Module mdl_Pro
  Public Function link()
    link = "dsn=dsDAUN;uid=;pwd="
  End Function
End Module
```

2. frmLogin

```
Public Class frmLogin

  Private Sub bersih()
    txtNama.Clear()
    txtPass.Clear()
    txtNama.Focus()
  End Sub

  Private Sub frmLogin_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call bersih()
  End Sub

  Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBatal.Click
    End
  End Sub

  Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogin.Click
    Try
      If txtNama.Text = "" Or txtPass.Text = "" Then
        MsgBox("Data Belum Lengkap, Silahkan Lengkapi Dahulu...", MsgBoxStyle.OkOnly +
        MsgBoxStyle.Question, "Konfirmasi Log In")
        Call bersih()
      Else
        Try
          Dim con As New System.Data.Odbc.OdbcConnection
          Dim cmd As System.Data.Odbc.OdbcCommand
          Dim dr As System.Data.Odbc.OdbcDataReader

          con.ConnectionString = mdl_Pro.link
          con.Open()
          cmd = New Odbc.OdbcCommand("select * from tblogin where nama_user=" & txtNama.Text &
          "" and pass_user=" & txtPass.Text & """, con)
          dr = cmd.ExecuteReader
          dr.Read()
          If Not dr.HasRows Then
            Call bersih()
          Else
            Me.Visible = False
            frmMenuUtama.Show()
          End If
          dr.Close()
          con.Close()
        Catch x As Exception
          MsgBox("Terjadi kesalahan pada [OK_Click] dengan pesan: " & Chr(13) & x.Message,
          MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
        End Try
      End If
    End Try
  End Sub
```

```

    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [OK_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint
    End Sub
End Class

```

3. frmMenuUtama

Imports System.Windows.Forms

Public Class frmMenuUtama

```

Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)

```

```

    ' Create a new instance of the child form.

```

```

    Dim ChildForm As New System.Windows.Forms.Form

```

```

    ' Make it a child of this MDI form before showing it.

```

```

    ChildForm.MdiParent = Me

```

```

    m_ChildFormNumber += 1

```

```

    ChildForm.Text = "Window " & m_ChildFormNumber

```

```

    ChildForm.Show()

```

```

End Sub

```

```

Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

```

```

    Dim OpenFileDialog As New OpenFileDialog

```

```

    OpenFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

```

```

    OpenFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"

```

```

    If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

```

```

        Dim FileName As String = OpenFileDialog.FileName

```

```

        ' TODO: Add code here to open the file.

```

```

    End If

```

```

End Sub

```

```

Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)

```

```

    Dim SaveFileDialog As New SaveFileDialog

```

```

    SaveFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

```

```

    SaveFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"

```

```

    If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

```

```

        Dim FileName As String = SaveFileDialog.FileName

```

```

        ' TODO: Add code here to save the current contents of the form to a file.

```

```

    End If

```

```

End Sub

```

```

Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)

```

```

    Me.Close()

```

```

End Sub

```

```

Private Sub CutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)

```

```

    ' Use My.Computer.Clipboard to insert the selected text or images into the clipboard

```

```

End Sub

```

```

Private Sub CopyToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)

```

```

    ' Use My.Computer.Clipboard to insert the selected text or images into the clipboard

```

```

End Sub

Private Sub PasteToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    'Use My.Computer.Clipboard.GetText() or My.Computer.Clipboard.GetData to retrieve information
    from the clipboard.
End Sub

Private Sub CascadeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.Cascade)
End Sub

Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.TileVertical)
End Sub

Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.TileHorizontal)
End Sub

Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.LayoutMdi(MdiLayout.ArrangeIcons)
End Sub

Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
    'Close all child forms of the parent.
    For Each ChildForm As Form In Me.MdiChildren
        ChildForm.Close()
    Next
End Sub

Private m_ChildFormNumber As Integer

Private Sub KeluarToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles KeluarToolStripMenuItem.Click
    End
End Sub

Private Sub DataKecamatanToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataKecamatanToolStripMenuItem.Click
    frmEntriKecamatan.Show()
End Sub

Private Sub DataWaliNagariToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataWaliNagariToolStripMenuItem.Click
    frmEntriWaliNagari.Show()
End Sub

Private Sub DataNagariToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataNagariToolStripMenuItem.Click
    frmEntriNagari.Show()
End Sub

Private Sub DataDAUNToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataDAUNToolStripMenuItem.Click
    frmTransaksiDAUN.Show()
End Sub

Private Sub DataDAUNToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataDAUNToolStripMenuItem1.Click
    frmLaporanDAUN.Show()
End Sub

```

```

Private Sub DataKecamatanToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataKecamatanToolStripMenuItem1.Click
    frmLaporanKecamatan.Show()
End Sub

```

```

Private Sub DataNagariToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataNagariToolStripMenuItem1.Click
    frmLaporanNagari.Show()
End Sub

```

```

Private Sub DataWaliNagariToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DataWaliNagariToolStripMenuItem1.Click
    frmLaporanWaliNagari.Show()
End Sub

```

```

Private Sub BantuanLogInToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BantuanLogInToolStripMenuItem.Click
    frmBantuanLogin.Show()
End Sub
End Class

```

4. frmEntriKecamatan

```
Public Class frmEntriKecamatan
```

```

Private Sub Bersih()
    cmbKodeKecamatan.Text = ""
    txtNamaKecamatan.Clear()
    Call IsiKodeKecamatan()
    cmbKodeKecamatan.Focus()
End Sub

```

```

Private Sub JumlahData()
    ListData.Refresh()
    lblByk.Text = ListData.Items.Count
End Sub

```

```

Private Sub IsiKodeKecamatan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct kd_kec from tbKecamatan ORDER BY kd_kec",
con)
        dr = cmd.ExecuteReader
        cmbKodeKecamatan.Items.Clear()
        While dr.Read
            cmbKodeKecamatan.Items.Add(dr.Item("kd_kec"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiKodeKecamatan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

Private Sub IsiDataKecamatan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbKecamatan where kd_kec=" &
cmbKodeKecamatan.Text & "", con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            txtNamaKecamatan.Text = dr.Item("kecamatan")
        Else
            txtNamaKecamatan.Clear()
        End If
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiDataKecamatan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

Private Sub Data()
    ListData.Columns.Clear()
    ListData.Items.Clear()
    ListData.Columns.Add("Kode Kecamatan")
    ListData.Columns(0).Width = 150
    ListData.Columns.Add("Nama Kecamatan")
    ListData.Columns(1).Width = 350
    Call isiList()
End Sub

```

```

Private Sub isiList()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbKecamatan ORDER BY kd_kec", con)
        dr = cmd.ExecuteReader
        ListData.Items.Clear()
        While dr.Read
            Dim satu = ListData.Items.Add(dr.Item("kd_kec"))
            satu.SubItems.Add(dr.Item("kecamatan"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [isiList()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    Call JumlahData()
End Sub

```

```

Private Sub frmEntriDataKecamatan_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Bersih()

```



```

        Call Data()
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnExit.Click
        Me.Close()
    End Sub

    Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnBatal.Click
        Call Bersih()
        Call Data()
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnSimpan.Click
        Try
            If cmbKodeKecamatan.Text = "" Then
                MsgBox("Kode Kecamatan Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                "Konfirmasi Simpan")
                Call Bersih()
            Else
                Try
                    Dim con As New System.Data.Odbc.OdbcConnection
                    Dim cmd As System.Data.Odbc.OdbcCommand
                    Dim dr As System.Data.Odbc.OdbcDataReader
                    Dim strSQL As String

                    con.ConnectionString = mdl_Pro.link
                    con.Open()
                    cmd = New Odbc.OdbcCommand("select * from tbKecamatan where kd_kec=" &
                    cmbKodeKecamatan.Text & "", con)
                    dr = cmd.ExecuteReader
                    If dr.Read Then
                        If MsgBox("Data Sudah Disimpan Sebelumnya, Apakah Ingin Di Edit???",
                        MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Konfirmasi Edit") = MsgBoxResult.Yes Then
                            strSQL = "update tbKecamatan set kecamatan=" & txtNamaKecamatan.Text & " where
                            kd_kec=" & cmbKodeKecamatan.Text & ""
                            MsgBox("Data Berhasil Di Edit...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                            "Konfirmasi Edit")
                            Call Bersih()
                            Call Data()
                        Else
                            cmbKodeKecamatan.Focus()
                            Exit Sub
                        End If
                    Else
                        strSQL = "insert into tbKecamatan (kd_kec,kecamatan) values (" & cmbKodeKecamatan.Text
                        & "," & txtNamaKecamatan.Text & ")"
                        MsgBox("Data Berhasil Di Simpan...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                        "Konfirmasi Simpan")
                        Call Bersih()
                        Call Data()
                    End If
                    dr.Close()
                    cmd.CommandText = strSQL
                    cmd.ExecuteNonQuery()
                    con.Close()
                Catch x As Exception
                    MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
                    MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
                End Try
            End Try
        End Sub

```

```

        End Try
    End If
    Call Bersih()
    Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
        MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnHapus.Click
    Try
        If cmbKodeKecamatan.Text = "" Then
            MsgBox("Kode Kecamatan Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
            "Konfirmasi Hapus")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand
                Dim dr As System.Data.Odbc.OdbcDataReader
                Dim strSQL As String

                con.ConnectionString = mdl_Pro.link
                con.Open()
                cmd = New Odbc.OdbcCommand("select * from tbKecamatan where kd_kec=" &
                cmbKodeKecamatan.Text & "", con)
                dr = cmd.ExecuteReader
                If dr.Read Then
                    If MsgBox("Penghapusan Data [ " & dr.Item("kd_kec") & " ] adalah Permanen !!!," & Chr(13)
                    & Chr(13) & "Yakin Akan Menghapus?", MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Konfirmasi
                    Hapus") = MsgBoxResult.Yes Then
                        strSQL = "delete from tbKecamatan where kd_kec=" & cmbKodeKecamatan.Text & ""
                        MsgBox("Data Berhasil Di Hapus...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                        "Konfirmasi Hapus")
                        Call Bersih()
                        Call Data()
                    Else
                        cmbKodeKecamatan.Focus()
                        Exit Sub
                    End If
                Else
                    MsgBox("Data Yang Akan Dihapus Tidak Ditemukan...", MsgBoxStyle.OkOnly +
                    MsgBoxStyle.Information, "Konfirmasi Hapus")
                    Call Bersih()
                    Call Data()
                    Exit Sub
                End If
                dr.Close()
                cmd.CommandText = strSQL
                cmd.ExecuteScalar()
                con.Close()
            Catch x As Exception
                MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
                MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
            End Try
        End If
        Call Bersih()
        Call Data()
    Catch x As Exception

```

```

        MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

Private Sub cmbKodeKecamatan_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeKecamatan.TextChanged
    Call IsiDataKecamatan()
End Sub
End Class

```

5. frmEntriNagari

```
Public Class frmEntriNagari
```

```

Private Sub Bersih()
    cmbKodeNagari.Text = ""
    txtNamaNagari.Clear()
    cmbKodeWaliNagari.Text = ""
    Call IsiKodeNagari()
    Call IsiKodeWaliNagari()
    cmbKodeNagari.Focus()
End Sub

```

```

Private Sub JumlahData()
    ListData.Refresh()
    lblByk.Text = ListData.Items.Count
End Sub

```

```

Private Sub IsiKodeNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct kd_nag from tbNagari ORDER BY kd_nag", con)
        dr = cmd.ExecuteReader
        cmbKodeNagari.Items.Clear()
        While dr.Read
            cmbKodeNagari.Items.Add(dr.Item("kd_nag"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiKodeNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

Private Sub IsiDataNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbNagari where kd_nag='" & cmbKodeNagari.Text
& "'", con)

```

```

dr = cmd.ExecuteReader
If dr.Read Then
    txtNamaNagari.Text = dr.Item("nagari")
    cmbKodeWaliNagari.Text = dr.Item("kodewn")
Else
    txtNamaNagari.Clear()
    cmbKodeWaliNagari.Text = ""
End If
dr.Close()
con.Close()
Catch x As Exception
    MsgBox("Terjadi kesalahan pada [IsiDataNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
End Try
End Sub

Private Sub IsiKodeWaliNagari()
Try
    Dim con As New System.Data.Odbc.OdbcConnection
    Dim cmd As System.Data.Odbc.OdbcCommand
    Dim dr As System.Data.Odbc.OdbcDataReader

    con.ConnectionString = mdl_Pro.link
    con.Open()
    cmd = New Odbc.OdbcCommand("select distinct kodewn from tbwn ORDER BY kodewn", con)
    dr = cmd.ExecuteReader
    cmbKodeWaliNagari.Items.Clear()
    While dr.Read
        cmbKodeWaliNagari.Items.Add(dr.Item("kodewn"))
    End While
    dr.Close()
    con.Close()
Catch x As Exception
    MsgBox("Terjadi kesalahan pada [IsiKodeWaliNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
End Try
End Sub

Private Sub IsiDataWaliNagari()
Try
    Dim con As New System.Data.Odbc.OdbcConnection
    Dim cmd As System.Data.Odbc.OdbcCommand
    Dim dr As System.Data.Odbc.OdbcDataReader

    con.ConnectionString = mdl_Pro.link
    con.Open()
    cmd = New Odbc.OdbcCommand("select * from tbWN where kodewn=" &
cmbKodeWaliNagari.Text & "", con)
    dr = cmd.ExecuteReader
    If dr.Read Then
        txtNamaWaliNagari.Text = dr.Item("NamaWN")
    Else
        txtNamaWaliNagari.Clear()
    End If
    dr.Close()
    con.Close()
Catch x As Exception
    MsgBox("Terjadi kesalahan pada [IsiDataWaliNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
End Try
End Sub

```

```

Private Sub Data()
    ListData.Columns.Clear()
    ListData.Items.Clear()
    ListData.Columns.Add("Kode Nagari")
    ListData.Columns(0).Width = 150
    ListData.Columns.Add("Nama Nagari")
    ListData.Columns(1).Width = 350
    ListData.Columns.Add("Kode Wali Nagari")
    ListData.Columns(2).Width = 150
    Call isiList()
End Sub

Private Sub isiList()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbNagari ORDER BY kd_nag", con)
        dr = cmd.ExecuteReader
        ListData.Items.Clear()
        While dr.Read
            Dim satu = ListData.Items.Add(dr.Item("kd_nag"))
            satu.SubItems.Add(dr.Item("nagari"))
            satu.SubItems.Add(dr.Item("kodewn"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [isiList()] dengan pesan: " & Chr(13) & x.Message,
        MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    Call JumlahData()
End Sub

Private Sub frmEntriDataKecamatan_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Bersih()
    Call Data()
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
    Me.Close()
End Sub

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click
    Call Bersih()
    Call Data()
End Sub

Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSimpan.Click
    Try
        If cmbKodeNagari.Text = "" Then
            MsgBox("Kode Nagari Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
            "Konfirmasi Simpan")
        End If
    End Try

```

```

        Call Bersih()
    Else
        Try
            Dim con As New System.Data.Odbc.OdbcConnection
            Dim cmd As System.Data.Odbc.OdbcCommand
            Dim dr As System.Data.Odbc.OdbcDataReader
            Dim strSQL As String
            con.ConnectionString = mdl_Pro.link
            con.Open()
            cmd = New Odbc.OdbcCommand("select * from tbnagari where kd_nag=" &
cmbKodeNagari.Text & "", con)
            dr = cmd.ExecuteReader
            If dr.Read Then
                If MsgBox("Data Sudah Disimpan Sebelumnya, Apakah Ingin Di Edit???",
MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Konfirmasi Edit") = MsgBoxResult.Yes Then
                    strSQL = "update tbnagari set nagari=" & txtNamaNagari.Text & ", kodewn=" &
cmbKodeWaliNagari.Text & " where kd_nag=" & cmbKodeNagari.Text & ""
                    MsgBox("Data Berhasil Di Edit...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Edit")
                    Call Bersih()
                    Call Data()
                Else
                    cmbKodeNagari.Focus()
                    Exit Sub
                End If
            Else
                strSQL = "insert into tbNagari (kd_nag,nagari,kodewn) values (" & cmbKodeNagari.Text &
", " & txtNamaNagari.Text & ", " & cmbKodeWaliNagari.Text & ""
                MsgBox("Data Berhasil Di Simpan...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Simpan")
                Call Bersih()
                Call Data()
            End If
            dr.Close()
            cmd.CommandText = strSQL
            cmd.ExecuteNonQuery()
            con.Close()
        Catch x As Exception
            MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
        End Try
    End If
    Call Bersih()
    Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnHapus.Click
    Try
        If cmbKodeNagari.Text = "" Then
            MsgBox("Kode Nagari Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Hapus")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand

```

```

Dim dr As System.Data.Odbc.OdbcDataReader
Dim strSQL As String

con.ConnectionString = mdl_Pro.link
con.Open()
cmd = New Odbc.OdbcCommand("select * from tbnagari where kd_nag=" &
cmbKodeNagari.Text & """, con)
dr = cmd.ExecuteReader
If dr.Read Then
    If MsgBox("Penghapusan Data [ " & dr.Item("kd_nag") & " ] adalah Permanen !!!," &
Chr(13) & Chr(13) & "Yakin Akan Menghapus?", MsgBoxStyle.YesNo + MsgBoxStyle.Information,
"Konfirmasi Hapus") = MsgBoxResult.Yes Then
        strSQL = "delete from tbnagari where kd_nag=" & cmbKodeNagari.Text & """
        MsgBox("Data Berhasil Di Hapus...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Hapus")
        Call Bersih()
        Call Data()
    Else
        cmbKodeNagari.Focus()
        Exit Sub
    End If
Else
    MsgBox("Data Yang Akan Dihapus Tidak Ditemukan...", MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "Konfirmasi Hapus")
    Call Bersih()
    Call Data()
    Exit Sub
End If
dr.Close()
cmd.CommandText = strSQL
cmd.ExecuteScalar()
con.Close()
Catch x As Exception
    MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
End Try
End If
Call Bersih()
Call Data()
Catch x As Exception
    MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
End Try
End Sub

Private Sub cmbKodeNagari_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeNagari.TextChanged
    Call IsiDataNagari()
End Sub

Private Sub cmbKodeWaliNagari_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeWaliNagari.TextChanged
    Call IsiDataWaliNagari()
End Sub
End Class

```

6. frmEntriWaliNagari

```
Public Class frmEntriWaliNagari
```

```
    Private Sub Bersih()
```

```

        cmbKodeWaliNagari.Text = ""
        txtNamaWaliNagari.Clear()
        cmbJenisKelamin.Text = ""
        txtAlamat.Clear()
        txtNoTelp.Clear()
        Call IsiKodeWaliNagari()
        cmbKodeWaliNagari.Focus()
    End Sub

Private Sub JumlahData()
    ListData.Refresh()
    lblByk.Text = ListData.Items.Count
End Sub

Private Sub IsiKodeWaliNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct kodewn from tbWN ORDER BY kodewn", con)
        dr = cmd.ExecuteReader
        cmbKodeWaliNagari.Items.Clear()
        While dr.Read
            cmbKodeWaliNagari.Items.Add(dr.Item("kodewn"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiKodeWaliNagari()] dengan pesan: " & Chr(13) & x.Message,
        MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiDataWaliNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbWN where kodewn="" &
        cmbKodeWaliNagari.Text & """, con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            txtNamaWaliNagari.Text = dr.Item("NamaWN")
            cmbJenisKelamin.Text = dr.Item("jekel")
            txtAlamat.Text = dr.Item("alamat")
            txtNoTelp.Text = dr.Item("notelp")
        Else
            txtNamaWaliNagari.Clear()
            cmbJenisKelamin.Text = ""
            txtAlamat.Clear()
            txtNoTelp.Clear()
        End If
        dr.Close()
        con.Close()
    Catch x As Exception

```



```

        MsgBox("Terjadi kesalahan pada [IsiDataWaliNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

Private Sub Data()
    ListData.Columns.Clear()
    ListData.Items.Clear()
    ListData.Columns.Add("Kode Wali Nagari")
    ListData.Columns(0).Width = 150
    ListData.Columns.Add("Nama Wali Nagari")
    ListData.Columns(1).Width = 300
    ListData.Columns.Add("Jenis Kelamin")
    ListData.Columns(2).Width = 100
    ListData.Columns.Add("Alamat")
    ListData.Columns(3).Width = 350
    ListData.Columns.Add("No Telp")
    ListData.Columns(4).Width = 100
    Call isiList()
End Sub

```

```

Private Sub isiList()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbwn ORDER BY kodewn", con)
        dr = cmd.ExecuteReader
        ListData.Items.Clear()
        While dr.Read
            Dim satu = ListData.Items.Add(dr.Item("kodewn"))
            satu.SubItems.Add(dr.Item("namawn"))
            satu.SubItems.Add(dr.Item("jkel"))
            satu.SubItems.Add(dr.Item("alamat"))
            satu.SubItems.Add(dr.Item("notelp"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [isiList()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    Call JumlahData()
End Sub

```

```

Private Sub frmEntriDataKecamatan_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Bersih()
    Call Data()
End Sub

```

```

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
    Me.Close()
End Sub

```

```

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click

```

```

    Call Bersih()
    Call Data()
End Sub

Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSimpan.Click
    Try
        If cmbKodeWaliNagari.Text = "" Then
            MsgBox("Kode Wali Nagari Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Simpan")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand
                Dim dr As System.Data.Odbc.OdbcDataReader
                Dim strSQL As String

                con.ConnectionString = mdl_Pro.link
                con.Open()
                cmd = New Odbc.OdbcCommand("select * from tbwn where kodewn="" &
cmbKodeWaliNagari.Text & """, con)
                dr = cmd.ExecuteReader
                If dr.Read Then
                    If MsgBox("Data Sudah Disimpan Sebelumnya, Apakah Ingin Di Edit???",
MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Konfirmasi Edit") = MsgBoxResult.Yes Then
                        strSQL = "update tbwn set namawn="" & txtNamaWaliNagari.Text & "", jekel="" &
cmbJenisKelamin.Text & "", alamat="" & txtAlamat.Text & "", notelp="" & txtNoTelp.Text & "" where
kodewn="" & cmbKodeWaliNagari.Text & ""
                        MsgBox("Data Berhasil Di Edit...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Edit")
                        Call Bersih()
                        Call Data()
                    Else
                        cmbKodeWaliNagari.Focus()
                        Exit Sub
                    End If
                Else
                    strSQL = "insert into tbwn (kodewn,namawn,jekel,alamat,notelp) values (" &
cmbKodeWaliNagari.Text & "," & txtNamaWaliNagari.Text & "," & cmbJenisKelamin.Text & "," &
txtAlamat.Text & "," & txtNoTelp.Text & ")"
                    MsgBox("Data Berhasil Di Simpan...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Simpan")
                    Call Bersih()
                    Call Data()
                End If
                dr.Close()
                cmd.CommandText = strSQL
                cmd.ExecuteNonQuery()
                con.Close()
            Catch x As Exception
                MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
            End Try
        End If
        Call Bersih()
        Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

```

```

End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnHapus.Click
    Try
        If cmbKodeWaliNagari.Text = "" Then
            MsgBox("Kode Wali Nagari Belum Di Isi...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Hapus")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand
                Dim dr As System.Data.Odbc.OdbcDataReader
                Dim strSQL As String

                con.ConnectionString = mdl_Pro.link
                con.Open()
                cmd = New Odbc.OdbcCommand("select * from tbwn where kodewn=" &
cmbKodeWaliNagari.Text & "", con)
                dr = cmd.ExecuteReader
                If dr.Read Then
                    If MsgBox("Penghapusan Data [ " & dr.Item("kodewn") & " ] adalah Permanen !!!," &
Chr(13) & Chr(13) & "Yakin Akan Menghapus?", MsgBoxStyle.YesNo + MsgBoxStyle.Information,
"Konfirmasi Hapus") = MsgBoxResult.Yes Then
                        strSQL = "delete from tbwn where kodewn=" & cmbKodeWaliNagari.Text & ""
                        MsgBox("Data Berhasil Di Hapus...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Hapus")
                        Call Bersih()
                        Call Data()
                    Else
                        cmbKodeWaliNagari.Focus()
                        Exit Sub
                    End If
                Else
                    MsgBox("Data Yang Akan Dihapus Tidak Ditemukan...", MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "Konfirmasi Hapus")
                    Call Bersih()
                    Call Data()
                    Exit Sub
                End If
                dr.Close()
                cmd.CommandText = strSQL
                cmd.ExecuteNonQuery()
                con.Close()
            Catch x As Exception
                MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
            End Try
        End Try
        Call Bersih()
        Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub cmbKodeWaliNagari_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeWaliNagari.TextChanged
    Call IsiDataWaliNagari()

```

```
End Sub
End Class
```

7. frmTransaksiDAUN

```
Public Class frmTransaksiDAUN
```

```
Private Sub Bersih()
    cmbNoKeputusan.Text = ""
    dtTanggal.Text = Date.Today
    cmbTahunDAUN.Text = ""
    cmbKodeKecamatan.Text = ""
    txtNamaKecamatan.Clear()
    cmbKodeNagari.Text = ""
    txtNamaNagari.Clear()
    txtJumlahDAUN.Clear()
    lblJumlahDAUN.Text = "Rp. " & Format(Val(txtJumlahDAUN.Text), "#,0.")
    Call IsiNoKeputusan()
    Call IsiTahunDAUN()
    Call IsiKodeKecamatan()
    Call IsiKodeNagari()
    cmbNoKeputusan.Focus()
End Sub

Private Sub JumlahData()
    ListData.Refresh()
    lblByk.Text = ListData.Items.Count
End Sub

Private Sub IsiNoKeputusan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct no_keputusan from tbtrans ORDER BY
no_keputusan", con)
        dr = cmd.ExecuteReader
        cmbNoKeputusan.Items.Clear()
        While dr.Read
            cmbNoKeputusan.Items.Add(dr.Item("no_keputusan"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiNoKeputusan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiDataKeputusan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
    End Try
End Sub
```

```

        cmd = New Odbc.OdbcCommand("select * from tbtrans where no_keputusan="" &
cmbNoKeputusan.Text & "" and kd_kec="" & cmbKodeKecamatan.Text & "" and kd_nag="" &
cmbKodeNagari.Text & """, con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            dtTanggal.Text = dr.Item("tanggal_keputusan")
            cmbTahunDAUN.Text = dr.Item("tahun_daun")
            txtJumlahDAUN.Text = dr.Item("jumlah_daun")
        Else
            dtTanggal.Text = Date.Today
            cmbTahunDAUN.Text = ""
            txtJumlahDAUN.Clear()
        End If
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiDataKeputusan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiKodeKecamatan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct kd_kec from tbKecamatan ORDER BY kd_kec",
con)
        dr = cmd.ExecuteReader
        cmbKodeKecamatan.Items.Clear()
        While dr.Read
            cmbKodeKecamatan.Items.Add(dr.Item("kd_kec"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiKodeKecamatan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiDataKecamatan()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbKecamatan where kd_kec="" &
cmbKodeKecamatan.Text & """, con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            txtNamaKecamatan.Text = dr.Item("kecamatan")
        Else
            txtNamaKecamatan.Clear()
        End If
        dr.Close()
    End Try
End Sub

```

```

        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiDataKecamatan()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiKodeNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select distinct kd_nag from tbNagari ORDER BY kd_nag", con)
        dr = cmd.ExecuteReader
        cmbKodeNagari.Items.Clear()
        While dr.Read
            cmbKodeNagari.Items.Add(dr.Item("kd_nag"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiKodeNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiDataNagari()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbNagari where kd_nag='" & cmbKodeNagari.Text
& "'", con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            txtNamaNagari.Text = dr.Item("nagari")
        Else
            txtNamaNagari.Clear()
        End If
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiDataNagari()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub IsiTahunDAUN()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()

```

```

        cmd = New Odbc.OdbcCommand("select distinct tahun_daun from tbtrans ORDER BY tahun_daun",
con)
        dr = cmd.ExecuteReader
        cmbTahunDAUN.Items.Clear()
        While dr.Read
            cmbTahunDAUN.Items.Add(dr.Item("tahun_daun"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiTahunDAUN()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub Data()
    ListData.Columns.Clear()
    ListData.Items.Clear()
    ListData.Columns.Add("No Keputusan")
    ListData.Columns(0).Width = 200
    ListData.Columns.Add("Tanggal Keputusan")
    ListData.Columns(1).Width = 100
    ListData.Columns.Add("Tahun DAUN")
    ListData.Columns(2).Width = 100
    ListData.Columns.Add("Kode Kecamatan")
    ListData.Columns(3).Width = 150
    ListData.Columns.Add("Kode Nagari")
    ListData.Columns(4).Width = 150
    ListData.Columns.Add("Jumlah DAUN")
    ListData.Columns(5).Width = 100
    ListData.Columns(5).TextAlign = HorizontalAlignment.Right
    Call isiList()
End Sub

Private Sub isiList()
    Try
        Dim con As New System.Data.Odbc.OdbcConnection
        Dim cmd As System.Data.Odbc.OdbcCommand
        Dim dr As System.Data.Odbc.OdbcDataReader

        con.ConnectionString = mdl_Pro.link
        con.Open()
        cmd = New Odbc.OdbcCommand("select * from tbtrans ORDER BY no_keputusan", con)
        dr = cmd.ExecuteReader
        ListData.Items.Clear()
        While dr.Read
            Dim satu = ListData.Items.Add(dr.Item("no_keputusan"))
            satu.SubItems.Add(dr.Item("tanggal_keputusan"))
            satu.SubItems.Add(dr.Item("tahun_daun"))
            satu.SubItems.Add(dr.Item("kd_kec"))
            satu.SubItems.Add(dr.Item("kd_nag"))
            satu.SubItems.Add(Format(dr.Item("jumlah_daun"), "#,0."))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [isiList()] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    Call JumlahData()
End Sub

```

```

Private Sub frmEntriDataKecamatan_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call Bersih()
    Call Data()
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
    Me.Close()
End Sub

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBatal.Click
    Call Bersih()
    Call Data()
End Sub

Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSimpan.Click
    Try
        If cmbNoKeputusan.Text = "" Or txtNamaKecamatan.Text = "" Or txtNamaNagari.Text = "" Then
            MsgBox("No Keputusan Atau Nama Kecamatan Atau Nama Nagari Belum Di Isi...",
                MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Konfirmasi Simpan")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand
                Dim dr As System.Data.Odbc.OdbcDataReader
                Dim strSQL As String

                con.ConnectionString = mdl_Pro.link
                con.Open()
                cmd = New Odbc.OdbcCommand("select * from tbtrans where no_keputusan=" &
                    cmbNoKeputusan.Text & " and kd_kec=" & cmbKodeKecamatan.Text & " and kd_nag=" &
                    cmbKodeNagari.Text & "", con)
                dr = cmd.ExecuteReader
                If dr.Read Then
                    If MsgBox("Data Sudah Disimpan Sebelumnya, Apakah Ingin Di Edit???",
                        MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Konfirmasi Edit") = MsgBoxResult.Yes Then
                        strSQL = "update tbtrans set tanggal_keputusan=" & dtTanggal.Value.ToString("yyyy-MM-dd") & ", tahun_daun=" & Val(cmbTahunDAUN.Text) & ", jumlah_daun=" &
                            Val(txtJumlahDAUN.Text) & " where no_keputusan=" & cmbNoKeputusan.Text & " and kd_kec=" &
                            cmbKodeKecamatan.Text & " and kd_nag=" & cmbKodeNagari.Text & ""
                        MsgBox("Data Berhasil Di Edit...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                            "Konfirmasi Edit")
                        Call Bersih()
                        Call Data()
                    Else
                        cmbNoKeputusan.Focus()
                    End If
                End If
            Else
                strSQL = "insert into tbtrans
                (no_Keputusan,tanggal_keputusan,tahun_daun,kd_kec,kd_nag,jumlah_daun) values (" &
                    cmbNoKeputusan.Text & "," & dtTanggal.Value.ToString("yyyy-MM-dd") & "," &
                    Val(cmbTahunDAUN.Text) & "," & cmbKodeKecamatan.Text & "," &
                    Val(txtJumlahDAUN.Text) & ")
                MsgBox("Data Berhasil Di Simpan...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
                    "Konfirmasi Simpan")
            End If
        End Try
    End Try

```



```

        Call Bersih()
        Call Data()
    End If
    dr.Close()
    cmd.CommandText = strSQL
    cmd.ExecuteNonQuery()
    con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    End If
    Call Bersih()
    Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnSimpan_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnHapus.Click
    Try
        If cmbNoKeputusan.Text = "" Or txtNamaKecamatan.Text = "" Or txtNamaNagari.Text = "" Then
            MsgBox("No Keputusan Atau Nama Kecamatan Atau Nama Nagari Belum Di Isi...",
MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Konfirmasi Hapus")
            Call Bersih()
        Else
            Try
                Dim con As New System.Data.Odbc.OdbcConnection
                Dim cmd As System.Data.Odbc.OdbcCommand
                Dim dr As System.Data.Odbc.OdbcDataReader
                Dim strSQL As String

                con.ConnectionString = mdl_Pro.link
                con.Open()
                cmd = New Odbc.OdbcCommand("select * from tbtrans where no_keputusan=" &
cmbNoKeputusan.Text & "" and kd_kec=" & cmbKodeKecamatan.Text & "" and kd_nag=" &
cmbKodeNagari.Text & """, con)
                dr = cmd.ExecuteReader
                If dr.Read Then
                    If MsgBox("Penghapusan Data [ " & dr.Item("no_keputusan") & " ] adalah Permanen !!!," &
Chr(13) & Chr(13) & "Yakin Akan Menghapus?", MsgBoxStyle.YesNo + MsgBoxStyle.Information,
"Konfirmasi Hapus") = MsgBoxResult.Yes Then
                        strSQL = "delete from tbtrans where no_keputusan=" & cmbNoKeputusan.Text & "" and
kd_kec=" & cmbKodeKecamatan.Text & "" and kd_nag=" & cmbKodeNagari.Text & ""
                        MsgBox("Data Berhasil Di Hapus...", MsgBoxStyle.OkOnly + MsgBoxStyle.Information,
"Konfirmasi Hapus")
                        Call Bersih()
                        Call Data()
                    Else
                        cmbNoKeputusan.Focus()
                    Exit Sub
                End If
            Else
                MsgBox("Data Yang Akan Dihapus Tidak Ditemukan...", MsgBoxStyle.OkOnly +
MsgBoxStyle.Information, "Konfirmasi Hapus")
                Call Bersih()
                Call Data()
            Exit Sub
        End If
    End If

```

```

        dr.Close()
        cmd.CommandText = strSQL
        cmd.ExecuteScalar()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
    End If
    Call Bersih()
    Call Data()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnHapus_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub cmbNoKeputusan_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbNoKeputusan.TextChanged
    Call IsiDataKeputusan()
End Sub

Private Sub cmbKodeKecamatan_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeKecamatan.TextChanged
    Call IsiDataKecamatan()
    Call IsiDataKeputusan()
End Sub

Private Sub cmbKodeNagari_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmbKodeNagari.TextChanged
    Call IsiDataNagari()
    Call IsiDataKeputusan()
End Sub

Private Sub txtJumlahDAUN_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtJumlahDAUN.TextChanged
    lblJumlahDAUN.Text = "Rp. " & Format(Val(txtJumlahDAUN.Text), "#,0.")
End Sub
End Class

```

8. frmLaporanDAUN

```
Public Class frmLaporanDAUN
```

```

    Private Sub Bersih()
        cmbNoKeputusan.Text = ""
        Call IsiNoKeputusan()
        cmbNoKeputusan.Focus()
    End Sub

    Private Sub IsiNoKeputusan()
        Try
            Dim con As New System.Data.Odbc.OdbcConnection
            Dim cmd As System.Data.Odbc.OdbcCommand
            Dim dr As System.Data.Odbc.OdbcDataReader

            con.ConnectionString = mdl_Pro.link
            con.Open()
            cmd = New Odbc.OdbcCommand("select distinct no_keputusan from tbtrans ORDER BY
no_keputusan", con)
            dr = cmd.ExecuteReader

```

```

        cmbNoKeputusan.Items.Clear()
        While dr.Read
            cmbNoKeputusan.Items.Add(dr.Item("no_keputusan"))
        End While
        dr.Close()
        con.Close()
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [IsiNoKeputusan()] dengan pesan: " & Chr(13) & x.Message,
        MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
    Me.Close()
End Sub

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click
    Call Bersih()
    crViewer1.RefreshReport()
End Sub

Private Sub frmLaporanDAUN_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Call Bersih()
End Sub

Private Sub btnCetak_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCetak.Click
    crViewer1.PrintReport()
End Sub

Private Sub btnTampil_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnTampil.Click
    Try
        crViewer1.ReportSource = Application.StartupPath & "\rptDataDAUN.rpt"
        crViewer1.SelectionFormula = "{tbTrans.no_keputusan}=" & cmbNoKeputusan.Text & ""
        crViewer1.RefreshReport()
        crViewer1.Zoom(75)
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnTampil_Click] dengan pesan: " & Chr(13) & x.Message,
        MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub
End Class

```

9. frmLaporanKecamatan

```

Public Class frmLaporanKecamatan

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
        Me.Close()
    End Sub

    Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click
        crViewer1.RefreshReport()
    End Sub

```

```

Private Sub btnCetak_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCetak.Click
    crViewer1.PrintReport()
End Sub

```

```

Private Sub btnTampil_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnTampil.Click
    Try
        crViewer1.ReportSource = Application.StartupPath & "\rptDataKecamatan.rpt"
        crViewer1.RefreshReport()
        crViewer1.Zoom(75)
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnTampil_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub
End Class

```

10. frmLaporanNagari

```

Public Class frmLaporanNagari

```

```

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
    Me.Close()
End Sub

```

```

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click
    crViewer1.RefreshReport()
End Sub

```

```

Private Sub btnCetak_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCetak.Click
    crViewer1.PrintReport()
End Sub

```

```

Private Sub btnTampil_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnTampil.Click
    Try
        crViewer1.ReportSource = Application.StartupPath & "\rptDataNagari.rpt"
        crViewer1.RefreshReport()
        crViewer1.Zoom(75)
    Catch x As Exception
        MsgBox("Terjadi kesalahan pada [btnTampil_Click] dengan pesan: " & Chr(13) & x.Message,
MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
    End Try
End Sub
End Class

```

11. frmLaporanWaliNagari

```

Public Class frmLaporanWaliNagari

```

```

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExit.Click
    Me.Close()
End Sub

```

```

Private Sub btnBatal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatal.Click

```

```
        crViewer1.RefreshReport()
    End Sub

    Private Sub btnCetak_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnCetak.Click
        crViewer1.PrintReport()
    End Sub

    Private Sub btnTampil_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnTampil.Click
        Try
            crViewer1.ReportSource = Application.StartupPath & "\rptDataWaliNagari.rpt"
            crViewer1.RefreshReport()
            crViewer1.Zoom(75)
        Catch x As Exception
            MsgBox("Terjadi kesalahan pada [btnTampil_Click] dengan pesan: " & Chr(13) & x.Message,
    MsgBoxStyle.Critical, "Kesalahan pada: " & Me.Text)
        End Try
    End Sub
End Class
```



KEMENTERIAN AGAMA
SEKOLAH TINGGI AGAMA ISLAM NEGERI BATUSANGKAR
PUSAT PENELITIAN DAN PENGABDIAN PADA MASYARAKAT

Jl. Sudirman No.137 Kuburajo Lima Kaum Batusangkar 27213 Telp. (0752) 71150, Ext 135, Fax. (0752) 71879
http://www.stainbatusangkar.ac.id e-mail: p3m.stainbatusangkar@gmail.com

Batusangkar, 13 November 2013

Nomor : St.02/IX/TL.00/ /87.0/2013
Lamp : 1 (satu) eksemplar
Hal : *Surat Rekomendasi Penelitian*

Kepada Yth.
Bupati Kabupaten Tanah Datar
Cq. Kepala Kantor KESBANGPOL Kabupaten Tanah Datar
di
Batusangkar

Assalamu 'alaikum Wr. Wb.
Dengan hormat,

Dengan ini disampaikan kepada Bapak/Ibu bahwa Mahasiswa yang namanya tersebut di bawah ini:

N a m a / NIM	: Robby Wahyudi / 10 205 059
Tempat/Tanggal Lahir	: Lima Kaum, 28 November 1991
Kartu Identitas	: KTP 1304042811910001
Jurusan	: Syariah
Program Studi	: Manajcmn Informatika
Alamat	: Jl. Sudirman No. 149 Jorong Kubu Rajo Nagari Lima Kaum Kecamatan Lima Kaum Kabupaten Tanah Datar

akan melakukan pengumpulan data untuk proses Penulisan Laporan Hasil Penelitiannya sebagai berikut:

J u d u l	: Desain Sistem Informasi Laporan Keuangan Dana Alokasi Umum Nagari (DAUN) pada Bagian Pemerintah Nagari dan Urusan Rantau di Pemerintahan Daerah Kabupaten Tanah Datar
Lokasi	: Kantor Pemerintahan Nagari
Waktu	: 13 November 2013 s/d 13 Januari 2014
Dosen Pembimbing 1	: Iswandi, M.Kom.
2	: -

untuk itu, diharapkan kiranya Bapak/Ibu berkenan memberi izin dalam rangka pelaksanaan penelitian mahasiswa yang bersangkutan.

Demikianlah disampaikan, atas bantuan dan kerjasamanya diucapkan terimakasih.

A.n Ketua,
Kepala P3M STAIN Batusangkar

ARDIMEN, M.Pd. Kons. †
NIP. 197205052001121002

Tembusan disampaikan kepada Yth.

1. Ketua STAIN Batusangkar (Sebagai Laporan)
2. Wakil Ketua Bidang Akademik dan Pengembangan Lembaga (Sebagai Laporan)
3. Ketua Jurusan Syariah
4. Ketua Program Studi Manajemen Informatika
5. Arsip



**SEKRETARIAT DAERAH KABUPATEN TANAH DATAR
BAGIAN PEMERINTAHAN NAGARI
DAN URUSAN RANTAU**

**Jln. Sultan Alam Bagagarsyah Telp. 71201 - 71301 Fax. 71201
BATUSANGKAR 27281**

SURAT KETERANGAN PENELITIAN

Dengan ini menyatakan nama yang tercantum dibawah ini telah melakukan penelitian/observasi pada Bagian Pemerintahan Nagari dan Urusan Rantau Setda Kabupaten Tanah Datar dengan baik pada tanggal 01 Januari sampai dengan 30 Januari 2014 dengan judul “ **Desain Sistem Informasi Penyalur Dana Alokasi Umum Nagari (DAUN)** ” guna untuk menyelesaikan tugas akhir pada Sekolah Tinggi Agama Islam Negeri Batusangkar.

NAMA : ROBBY WAHYUDI
NIM : 10 205 059
SEMESTER : VII (Tujuh)
JURUSAN : SYARIAH
PRODI : MANAJEMEN INFORMATIKA

Demikianlah disampaikan, atas perhatiannya diucapkan terimakasih.

**Plt. KEPALA BAGIAN
PEMERINTAHAN NAGARI DAN URUSAN RANTAU
KASUBAG PEMBERDAYAAN KELEMBAGAAN NAGARI**

**A. H. MIZA AZIZ, S.Sos
NIP. 19710912 200201 1 005**