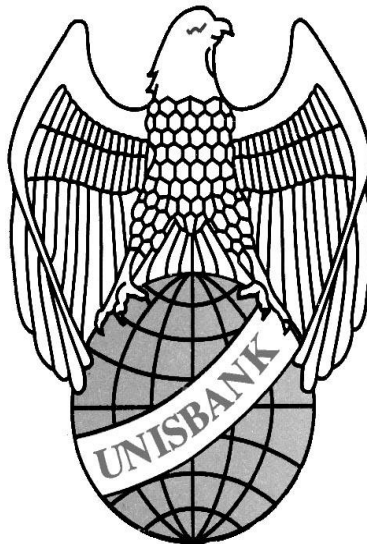


**SIMULASI SISTEM PENGGUNAAN RUANGAN BERDASARKAN
JADWAL DENGAN MENGGUNAKAN SENSOR SIDIK JARI
TUGAS AKHIR**

Tugas akhir disusun untuk memenuhi syarat
Mencapai gelar kesarjanaan komputer pada
Program Studi Teknik Informatika
Jenjang Program Strata-1



Oleh :
Hafit Rohman
08.01.53.0165
9953

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK)
SEMARANG
2013**

PERNYATAAN KESIAPAN UJIAN TUGAS AKHIR

Saya, Hafit Rohman, dengan ini menyatakan bahwa Laporan Tugas Akhir yang berjudul :

**SIMULASI SISTEM PENGGUNAAN RUANGAN BERDASARKANN
JADWAL DENGAN MENGGUNAKAN SENSOR SIDIK JARI** adalah benar hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain.

(Hafit Rohman)

NIM : 08.01.53.0165

Disetujui oleh Pembimbing

Kami setuju Laporan tersebut diajukan untuk Ujian Tugas Akhir

(Eddy Nurraharjo, ST, M.Cs)

Semarang : 20 February 2013

Pembimbing I

(Saefurrohman, S.Kom, M.Cs)

Semarang : 20 February 2013

Pembimbing II

HALAMAN PENGESAHAN

Telah dipertahankan di depan tim dosen penguji Tugas Akhir Fakultas Teknologi Informasi UNIVERSITAS STIKUBANK (UNISBANK) Semarang dan diterima sebagai salah satu syarat guna menyelesaikan Jenjang Program Strata-1, Program Studi : Teknik Informatika.

Semarang, 20 February 2013

Ketua

(Eddy Nurraharjo, ST, M.Cs)

Sekretaris

(Saefurrohman, S.Kom, M.Cs)

Anggota

(Zuly Budiarmo, IR, M.Cs)

MENGETAHUI:

UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Fakultas Teknologi Informasi
Dekan

(Dwi Agus Diartono, S.Kom, M.Kom)

MOTTO DAN PERSEMBAHAN

- ❖ Ilmu akan memanggil amal. Bila ilmu dijawab.
- ❖ Kegagalan adalah kunci keberhasilan.
- ❖ Suburkanlah keimanan dengan baja-baja ilmu.
- ❖ Jika seseorang tidak berusaha, padahal nasibnya telah mengharuskannya berusaha, dia menyia-nyiakan telah nasibnya itu, dan akan ditinggalkan. Namun orang yang bertekad baja tidak pernah menyerah pada ujian, akan selalu melihat masalah dengan mata terbuka. Dia adalah penembus zaman, yang selalu bergerak, jika ditutup satu pintu dia akan menerobos pintu yang lain.
- ❖ Jadilah engkau orang yang kakinya berada di tanah, namun cita-citanya menggantung di langit.
- ❖ Kesabaran salah satu kunci dari keberhasilan.

Kupersembahkan kepada :

Maha Suci Allah SWT

Keluarga terutama kedua orangtuaku ;

My lovely yang telah memberi semangat dan motivasi.

Serta semua orang yang telah memberi motivasi dan inspirasi.

FAKULTAS TEKNIK INFORMATIKA
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Program Studi : Teknik Informatika
Tugas Akhir Sarjana Komputer
Semester Ganjil Tahun 2013

**Simulasi Sistem Penggunaan Ruang Berdasarkan Jadwal Dengan
Menggunakan Sensor Sidik Jari**

(Hafit Rohman)

Nim : 08.01.53.0165

Abstraksi

Bersumber pada system informasi yang telah ada di lingkungan universitas stikubank semarang serta melihat adanya kekurangan yang ada maka penulis tertarik untuk melakukan suatu penelitian di universitas stikubank semarang.

Tujuan penelitian ini adalah untuk membuat sebuah system penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari. Dimana untuk masuk ruangan kelas berdasarkan jadwal harus meletakkan sidik jari mahasiswa maupun dosen di atas sensor sidik jari (fingerprint). kemudian aplikasi akan memproses penggunaan ruangan secara otomatis sesuai dengan jadwal yang sudah ada.

Penyusunan dalam membuat simulasi system penggunaan ruangan berdasarkan jadwal ini dengan menggunakan bahasa pemrograman Borland Delphi 6, dalam rancangan simulasi system penggunaan ruangan ini menggunakan diagram konteks, DFD, ERD (Entity Relationship Diagram).

Kata kunci : Simulasi System, Penggunaan Ruang, Fingerprint.

Semarang : 20 Februari 2013

Pembimbing I

Pembimbing II

Eddy Nurraharjo ST, M.Cs

Saefurrohman, S.Kom, M.Cs

KATA PENGANTAR

Alhamdulillah puji syukur saya panjatkan kehadirat Allah yang maha kuasa atas Rahmat, Taufiq, dan Hidayah-Nya sehingga pembuatan Tugas Akhir ini dapat diselesaikan tepat pada waktunya.

Pada Tugas Akhir ini saya memilih judul “SIMULASI SISTEM PENGGUNAAN RUANGAN BERDASARKAN JADWAL DENGAN MENGGUNAKAN SENSOR SIDIK JARI” ini merupakan tugas akhir yang saya susun untuk mencapai gelar sarjana komputer (S.Kom) pada Program Studi teknik Informatika, jenjang program strata-1 Universitas Stikubank (UNISBANK) Semarang. Dalam penyusunan tugas akhir ini, data saya ambil dari perpustakaan dan maupun internet, yang kemudian disajikan kembali dengan dasar pengetahuan yang telah saya peroleh selama ini.

Menyadari kemampuan yang terbatas pada saya, tentu saja laporan ini tidak dapat diselesaikan dengan baik tanpa bantuan dari berbagai pihak, oleh karena itu pada kesempatan ini pula, saya sampaikan terima kasih atas segala bantuan yang telah diberikan baik berupa bimbingan, dukungan, semangat, serta saran sehingga tugas akhir ini dapat diselesaikan dengan baik.

Ucapan terima kasih saya haturkan kepada :

1. Allah SWT yang selalu melimpahkan segala kemuliaan kepada umatnya;
2. Bapak dan Ibu yang kusayangi dan kucintai yang tak pernah henti untuk mendoakan dan selalu memberikan nasehat – nasehat yang dapat memberikan semangat didalam setiap langkahku;
3. Bapak Dr. Bambang Suko Priyono, M.M selaku Rektor Universitas Stikubank (UNISBANK) Semarang;

4. Bapak Dwi Agus Diartono, S.Kom, M.Kom selaku Dekan Teknik Informatika Universitas Stikubank Semarang;
5. Dewi Handahani, ST, M.KOM selaku Ketua Program Studi Teknik Informatika di Universitas Stikubank (UNISBANK) Semarang;
6. Bu Sri Eniyati, S.Kom, M.Cs selaku dosen wali yang banyak memberikan saran, pengarahan, dan bantuan penulis selama ini;
7. Eddy Nurraharjo, ST, M.Cs selaku dosen Pembimbing I yang membantu mengarahkan baik sistematika dan tata bahasa serta memberi saran – saran dalam penyusunan tugas akhir ini;
8. Saefurrohman, S.Kom, M.Cs selaku Pembimbing II yang telah membantu mengarahkan baik sistematika dan tata bahasa serta memberi saran-saran dalam penyusunan tugas akhir ini;
9. Bapak dan Ibu Dosen Pengajar Program Studi Teknik Informatika di Universitas Stikubank Semarang, atas segala bimbingan selama saya menempuh studi;
10. Semua teman-temanku 'Teknik Informatika' angkatan 2008 yang selalu memberikan semangat dan motivasi "thx sob, your best friends".

Sebagai manusia biasa, saya menyadari bahwa tugas akhir ini masih jauh dari sempurna, oleh karena itu segala kritik dan saran yang bersifat membangun menyempurnakan laporan ini sangat saya harapkan. Semoga hasil penelitian ini dapat bermanfaat bagi pembaca pada umumnya dan almamater pada khususnya.

Semarang, 20 February 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
ABSTRAKSI.....	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan dan Manfaat Tugas Akhir	4
1.5. Manfaat Penelitian	4
1.6 Metode Penelitian	5
1.6.1 Obyek Penelitian	5
1.6.2 Metode Pengumpulan Data	5
1.6.3 Tahapan Pengembangan Sistem.....	6
1.6. Sistematika Penulisan	8

BAB II TINJAUAN PUSTAKA

2.1. Analisis Dan Implementasi Algoritma Genetika Pada Penjadwalan Mata Kuliah	10
2.2. Pembuka Pintu Otomatis Menggunakan Avr Atmega 8535 Dan Sensor Pir.....	12
2.3. Prototipe Sistem Keamanan Rumah Menggunakan <i>Webcam Dan Finger Print</i> Berbasis <i>Web</i> Dan <i>Sms</i>	14
2.4. Integrasi Pengendalian Pintu Masuk Ruang Dengan Sistem Informasi Data Dengan Bantuan Alat Pembaca Barcode.....	15

BAB III LANDASAN TEORI

3.1 Mengenal Simulasi	18
3.2. Borland Delphi	19
3.3.1.Mengenal IDE Delphi.....	20
3.4. Analisis Sistem	26
3.4.1. Data Flow Diagram (DFD).....	26
3.4.2. Entity Relationship Diagram	27

BAB IV ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem	30
4.1.1. Narasi Proses Alir Dokumen Pada Sistem Lama.....	30
4.1.2. Kelebihan dan Kekurangan Pada Sistem Lama.....	31
4.1.3. Identifikasi Masalah Pada Sistem Lama.....	31
4.1.4. Identifikasi kebutuhan data dan informasi.....	32

4.1.5. Alternatif Sistem Yang Diusulkan.....	32
4.1.6. Identifikasi Kebutuhan Hardware.....	32
4.1.7. Identifikasi Kebutuhan Perangkat Lunak.....	34
4.2. Perancangan Sistem.....	35
4.2.1. Context Diagram.....	35
4.2.2. Diagram Decomposisi.....	36
4.2.3. DFD levelled	36.
4.2.4. ERD (<i>Entity Relationship Diagram</i>).....	38
4.2.5. Relationship Tabel.....	39
4.2.6. Data Dictionary (Kamus Data)	39
4.3. Perancangan Tampilan.....	45
4.3.1 Perancangan Input Output.....	46

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi	59
5.1.1. Eksekusi Simulasi Sistem.....	59
5.2. Pengujian Alat	79
5.2.1. Pengujian Power Supply	79
5.2.2 Pengujian Microcontroller	81
5.2.3 Pengujian Driver Motor Pintu.....	84
5.3 Pengujian Sistem Secara Keseluruhan.....	86

BAB VI PENUTUP

5.1. Kesimpulan.....	88
5.2. Saran.....	89

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel Matakuliah	39
Tabel 4.2 Tabel Mahasiswa.....	40
Tabel 4.3 Tabel Dosen	40
Tabel 4.4 Tabel Jadwal Kuliah.....	41
Tabel 4.5 Tabel Masuk Ruang	42
Tabel 5.1 Hasil dan analisa power supply.....	80
Tabel 5.2 Hasil dan analisa mikrokontroler	83
Tabel 5.3 Hasil dan analisa drive motor pintu	85

DAFTAR GAMBAR

	Halaman
Gambar 4.1. Context Diagram	35
Gambar 4.2. Diagram Decomposisi.....	36
Gambar 4.3. DFD Level 1 Pendataan	36
Gambar 4.4. DFD level 2 transaksi	37
Gambar 4.5. DFD level 3 laporan	37
Gambar 4.6. Entity Relationship Diagram.....	38
Gambar 4.7. Relationship Tabel.....	39
Gambar 5.1 Tampilan Login	59
Gambar 5.2 Tampilan Menu Utama	61
Gambar 5.3 Tampilan data dosen.....	62
Gambar 5.4 Tampilan data mahasiswa	63
Gambar 5.5 Tampilan Data matakuliah	64
Gambar 5.6 Tampilan Data jadwal.....	66
Gambar 5.7 Tampilan Data Masuk Ruangan.....	68
Gambar 5.8 Tampilan Edit Dosen	69
Gambar 5.9 Tampilan edit mahasiswa.....	72
Gambar 5.10 Tampilan edit matakuliah.....	74
Gambar 5.11 Tampilan edit jadwal	75

Gambar 5.12 Tampilan Absen Masuk Ruang	77
Gambar 5.13 Tampilan Ubah Password	77
Gambar 5.14 Tampilan set com	79
Gambar 5.15 Pengujian Power Supply	80
Gambar 5.16 Pengujian Mikrocontroller	82
Gambar 5.17 Pengujian Drive Motor Pintu	85
Gambar 5.18 Bagian Luar Simulasi Sistem Penggunaan Ruangan Berdasarkan Jadwal Dengan Menggunakan Sensor Sidik Jari...	86

DAFTAR LAMPIRAN

Lampiran A	Listing Program
Lampiran B	Lembar Bimbingan
Lampiran C	Gambar Rangkain Keseluruhan

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan ilmu pengetahuan dan teknologi dewasa ini sangat berkembang pesat, sehingga mempunyai arti penting dan sangat berpengaruh terhadap kehidupan manusia. Hal ini terlihat dengan adanya berbagai kemudahan yang diciptakan dan ditawarkan. Oleh karena itu dengan adanya kecanggihan teknologi ini, maka dibutuhkan sumber daya manusia yang tersedia dan harus disiapkan untuk dimanfaatkan agar sejalan dengan perkembangan yang ada, sehingga masyarakat tidak akan kekurangan sumber daya manusia untuk menciptakan teknologi dan tidak ketertinggalan zaman.

Diantara perkembangan-perkembangan teknologi yang sangat canggih di zaman sekarang ini salah satunya dapat kita lihat dibidang teknologi berbasis komputerisasi, yang memberikan suatu alat bantu didalam suatu sistem yang memudahkan aktifitas manusia pada umumnya. Salah satunya didalam penggunaan ruangan kelas berdasarkan jadwal yang sudah ditentukan waktu dan tempatnya.

Didalam sebuah lembaga pendidikan atau lembaga layanan publik lainnya yang telah memiliki sistem informasi data dapat menambahkan aplikasi pengendalian pintu masuk ruangan ruang dengan mengintegrasikannya ke sistem informasi data yang telah dimiliki dan dengan bantuan pembaca kode akses dengan cepat.

Sebenarnya terdapat banyak metode yang dapat digunakan sebagai pembatas akses masuk ruang seperti dengan menggunakan barcode, retina mata, suara, dan wajah. Alasan dalam pemilihan pembaca sidik jari sebagai pembatas akses adalah untuk kepentingan praktis karena setiap orang yang memiliki akses masuk merupakan anggota dari lembaga tersebut dan memiliki sidik jari yang berbeda-beda dan yang digunakan untuk memasuki ruangan agar dapat melakukan belajar dan mengajar, dan memanfaatkan fasilitas lain.

Didalam menggunakan ruangan kelas pada saat ini sangatlah sederhana didalam pengoperasiannya, hanya membuka dan menutup pintu secara manual, dan dilihat dari segi keamanannya sangatlah kurang efektif karena masih saja terjadi pencurian sewaktu-waktu, oleh karna kata itu perlu adanya sistem yang memberikan suatu kemudahan dan keamanan yang dapat dilakukan dan dipergunakan oleh orang yang bersangkutan sesuai jadwal penggunaan yang sudah ditentukan, agar tidak terjadi kesalah gunaan ruang tersebut, dan tidak sembarangan orang dapat keluar masuk ruangan kelas.

Oleh karena itu kita dapat menggunakan kode-kode dengan berbagai variasi kombinasi hanya guru-guru atau dosen matakuliah dan mahasiswa yang bersangkutan yang dikasih wewenang untuk mengakses sitem ini sesuai jadwal ruangan yang akan digunakan untuk belajar-mengajar diruangan.

Simulasi sistem penggunaan ruangan yang akan dijalankan keseluruhannya diakses dengan menggunakan kode-kode yang dapat diwujudkan dengan salah satunya menggunakan kombinasi--kombinasi sidik jari yang dimiliki bagi setiap dosen atau orang yang mengajar sesuai jadwal

yang sudah ditentukan, dengan menggunakan sidik jari seseorang yang tidak bersangkutan tidak dapat mengakses sistem tersebut. Sehingga perlu adanya simulasi sistem yang seperti ini agar dapat mencegah terjadinya kesalahan penggunaan ruangan tersebut, membantu kelancaran dan kemudahan pembelajaran dikelas dan sekaligus menjadi judul skripsi saya, yaitu : “**Simulasi Sistem Penggunaan Ruang Berdasarkan Jadwal Dengan Menggunakan Sensor Sidik Jari**”.

1.2 Perumusan masalah

Berdasarkan latar belakang seperti diatas yang akan menjadi pokok permasalahan dalam laporan tugas akhir ini adalah bagaimana merancang suatu simulasi sistem buka tutup pintu ruang berdasarkan jadwal penggunaan ruangan dengan menggunakan sensor sidik jari.

1.3 Pembatasan Masalah

Dalam penulisan laporan tugas akhir ini, peneliti membatasi permasalahan yang diambil dan hal ini dikarenakan agar tidak menyimpang dan melebar luas dari permasalahan yang sedang dihadapi. Maka batasan masalah tersebut adalah meliputi :

- a. Sistem hanya dapat menerima input kode-kode dari angka, huruf ataupun bisa dengan sensor sidik jari.
- b. Sistem hanya mengujian simulasi sistem penggunaan 3 ruang kelas, 3 dosen pengampu dan 6 mahasiswa berdasarkan jadwal yang sudah ada.
- c. Sistem menggunakan Bahasa Pemrograman Borland Delphi 6.0

1.4 Tujuan Penelitian

Tujuan umum penyusunan tugas akhir ini adalah merancang suatu simulasi sistem penggunaan ruangan berdasarkan jadwal yang dapat diakses dengan menggunakan sensor sidik jari. Sedangkan tujuan khususnya adalah merancang simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari yang nantinya akan dapat memberikan suatu kemudahan bagi pengguna ruangan, dan tidak ada kesalahan dalam memasuki ruangan kelas.

1.5 Manfaat Penelitian

Adapun manfaat yang akan dapat diperoleh dari penyusunan tugas akhir ini adalah sebagai berikut :

a. Bagi Akademik :

Dapat digunakan sebagai bahan informasi , referensi, artikel dan masukan positif untuk bahan pembelajaran bagi mahasiswa, serta menambah perbendaharaan literature bagi perpustakaan di Universitas Stikubank Semarang.

b. Bagi Penulis :

Sebagai sarana menerapkan dan mengembangkan ilmu yang diperoleh pada saat diperkuliahan sehingga dapat diterapkan didalam dunia kerja.

c. Bagi pengguna :

Memberikan suatu kemudahan dan keamanan didalam penggunaan ruangan sesuai jadwal yang sudah ditentukan oleh BAAK

1.6 Metode Penelitian

1.6.1 Obyek Penelitian

Obyek penelitian ini adalah di ruangan Universitas Stikubank Semarang

1.6.2 Metode pengumpulan data

1. Studi Lapangan

Yaitu mengadakan *survey* langsung terhadap obyek penelitian dengan cara :

a. Wawancara

Mengadakan wawancara secara langsung terhadap obyek penelitian untuk memperoleh data yang dibutuhkan.

b. Observasi

Mengadakan pengamatan secara langsung terhadap obyek penelitian.

c. Dokumentasi

Mengumpulkan data dengan membaca arsip-arsip atau file-file yang ada pada obyek penelitian.

d. Studi Pustaka

Dimaksudkan untuk menggali teori yang berhubungan dengan penulisan laporan ini dan merupakan tahap awal dari pelaksanaan Tugas Akhir ini.

Dari hasil pengumpulan data tersebut diharapkan dapat diperoleh data sebagai berikut :

1. Data primer

Data primer adalah suatu data yang diperoleh secara langsung dari sumber atau yang menjadi objek penelitian yang sudah jadi dan siap dipakai.

2. Data primer

Data primer adalah data yang diperoleh dari berbagai sumber buku-buku atau dokumentasi yang ada kaitannya, maka dari itu data ini digunakan sebagai perlengkapan untuk menganalisa dan merancang masalah ini.

1.6.3 Tahapan Pengembangan Sistem

Dalam menganalisa dan merancang simulasi sistem penggunaan ruangan dengan menggunakan sensor sidik jari ini lebih mengacu pada metode pengembangan sistem yang menggunakan sistem SDLC (*system development life cycle*).

Adapun tahapan-tahapan yang harus dilakukan dalam menggunakan metode pengembangan sistem SDLC (*system development life cycle*), adalah sebagai berikut :

1. Tahap Perancangan

Mengenali, mendiagnosa dan mendefinisikan masalah yang diterima dari pemakai :

Kegiatan yang akan dilakukan adalah sebagai berikut :

- a. Mendefinisikan masalah, tujuan pokok permasalahan
- b. Mempelajari struktur organisasi dan fungsi organisasi

- c. Menentukan prioritas penanganan masalah
- d. Merekomendasikan sistem pendekatan yang akan digunakan dalam pemecahan masalah ini

2. Tahap Analisis

Melakukan dan mengadakan studi kelayakan terhadap organisasi pemakai dan melibatkan sistem lebih rinci dalam kegiatan yang akan dilakukan untuk menganalisis adalah sebagai berikut:

- a. Studi pustaka
- b. Mempelajari kembali struktur organisasi serta aliran sumber-sumber dan narasumber yang telah didapat
- c. Mengembangkan alternative pemecahan masalah yang telah ditentukan.

3. Tahap Desain

Menyusun sistem dan menuangkan secara tertulis, alur data beserta sumber informasinya bertujuan untuk mendesain sistem yang baru dan didapat menyelesaikan masalah-masalah yang akan dihadapi oleh pengguna simulasi sistem jadwal penggunaan ruang dengan menggunakan sensor sidik jari

4. Tahap Implementasi

Sasaran dan tujuan tahapan ini adalah untuk mempelajari dan efektifitas sistem yang baru, serta melakukan pemeliharaan dan evaluasi sistem yang akan diterapkan.

1.7 Sistematika penulisan

Secara garis besar dalam penulisan tugas akhir ini dibagi menjadi 6 bab yang tersidi sebagai berikut, yaitu :

Bab I : Pendahuluan

Berisikan tentang latar belakang masalah, rumusan dan batasan masalah, dan mempunyai manfaat dan tujuan penelitian, metode penelitian, dan sistematika penulisan tugas akhir ini.

Bab II : Landasan Teori

Didalam landasan teori ini berisikan tentang kupasan butir-butir kutipan dari berbagai referensi baik yang ilmiah maupun non ilmiah. Kutipan-kutipan pustaka tersebut biasanya punya hubungan dengan masalah yang sedang akan diteliti Kutipan-kutipan itu bisa berasal dari buku-buku teks ilmiah, jurnal ilmiah, internet.

Bab III : Analisa dan Perancangan Sistem

Membuat analisis aliran kerja manajemen sistem yang sedang berjalan dan menggambarkan kerangka perancangan sistem secara alur diagram. Kerangka ini sekaligus memperlihatkan pola pikir yang akan dilakukan ketika akan memulai dan mengakhiri suatu kegiatan penelitian.

Tahap demi tahap penelitian yang tergambar dalam alur diagram itu diuraikan secara singkat dan jelas misalnya, apa dan bagaimana tahap yang akan dilakukan.

Bab IV: Implementasi Sistem

Pada bagian ini berisikan pengujian system yang hasil implementasi yang telah dirancang, apakah berhasil atau gagal dan bukan menjelaskan bagaimana menggunakannya. Jika hasil implementasi system belum baik, maka harus dicari penyebabnya hingga bisa berjalan sukses.

Bab V: kesimpulan dan saran

Pada bagian ini berisikan tentang kesimpulan, saran, Petunjuk teknis pembuatan laporan tugas akhir ini dan membuat laporan hasil penelitian.

BAB II

TINJAUAN PUSTAKA

2.1. ANALISIS DAN IMPLEMENTASI ALGORITMA GENETIKA PADA PENJADWALAN MATA KULIAH

Abdullah (2010), menyampaikan Proses penjadwalan kuliah adalah suatu proses untuk memasangkan event yang berisi komponen matakuliah dan dosen pada timeslot yang berisi komponen waktu dan ruang. Hasil penjadwalan ini berisi susunan jadwal yang tidak melanggar syarat hard constraint atau jadwal tidak saling bertubrukan satu sama lain. Selain itu hasil jadwal juga diharapkan dapat meminimumkan pelanggaran soft constraint.

Penetapan soft constraint biasanya digunakan untuk membuat hasil jadwal lebih baik. Semakin sedikit jumlah pelanggaran soft constraint maka hasil penjadwalan semakin sesuai dengan harapan. Untuk memecahkan permasalahan penjadwalan kuliah ini penulis menggunakan algoritma genetika. Algoritma ini didasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Pada algoritma genetika dibuat kromosom-kromosom dari populasi awal secara random, sehingga memungkinkan kromosom tersebut berkembang biak sesuai hukum-hukum evolusi yang nantinya diharapkan akan dapat menghasilkan kromosom dengan nilai yang lebih baik dari kromosom-kromosom yang ada pada populasi awal.

Kromosom merupakan representasi solusi dari permasalahan yang diangkat, sehingga apabila kromosom yang baik tersebut dihasilkan, maka diharapkan solusi yang baik dari permasalahan jadwal juga didapatkan. Berdasarkan pengujian yang telah dilakukan, dengan melakukan perubahan parameter masukan, maka akan memberikan hasil yang bervariasi. Kemudian setelah dilakukan pengujian terhadap parameter, maka didapatkan parameter masukan yang dapat menghasilkan jadwal terbaik dengan pelanggaran terhadap *soft constraint* bernilai 0, yaitu jumlah kromosom 20, probabilitas mutasi 50%, dan probabilitas cross over 50%, yang dihasilkan sebelum generasi ke-100.

Berdasarkan hasil yang didapat dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuannya, maka diperoleh kesimpulan sebagai berikut:

- 1) Algoritma genetika bisa diterapkan untuk menyelesaikan permasalahan yang biasanya terjadi pada saat penyusunan jadwal mata kuliah. Kemudian apabila dilakukan analisis yang lebih detail maka kemungkinan untuk mengatasi semua permasalahan terutama yang ada pada batasan masalah bisa dilakukan.
- 2) Dengan melakukan perubahan parameter masukan, maka akan memberikan hasil yang bervariasi. Kemudian setelah dilakukan pengujian terhadap parameter, maka didapatkan parameter masukan yang dapat menghasilkan jadwal terbaik dengan pelanggaran terhadap *soft constraint* yang bernilai 0.

2.2. PEMBUKA PINTU OTOMATIS MENGGUNAKAN AVR ATMEGA 8535 dan SENSOR PIR

Ebiezer, Supriyanto (2010), yang mengemukakan dengan kemajuan teknologi seperti sekarang penulis membuat alat yang berfungsi untuk mengamankan kondisi dalam rumah atau ruangan. Penulis mencoba membuat modul keamanan pintu yang berbasis mikrokontroler avr atmega 8535. Tujuan pembuatan system ini untuk memberikan alternatif pengamanan. Sistem ini berfungsi sebagai pengganti kunci konvensional.

Sistem pintu satu dapat berinteraksi dengan sistem pintu yang lainnya, pada masing-masing sistem pintu dapat me-remote sistem pintu lainnya. Sistem keamanan pintu ini dapat memberikan keamanan karena alat ini berfungsi jika password yang dimasukkan melalui keypad dalam keadaan benar, maka sensor pir akan mendeteksi keberadaan orang di depan pintu akan terbuka secara otomatis dan akan tertutup secara otomatis. Jika sensor tidak mendeteksi keberadaan orang di depan pintu maka pintu tetap tertutup. Jika password yang dimasukkan salah maka akan ada peringatan berupa buzzer akan berbunyi.

Sistem ini terdiri atas perangkat lunak dan perangkat keras. perangkat keras terdiri dari sensor PIR dan keypad sebagai inputan perangkat lunak terdiri dari bahasa C yang diisikan dalam mikrokontroler sebagai pemroses data. LCD akan memberikan informasi berupa karakter tentang langkah- langkah keamanan pintu, motor dc berfungsi untuk membuka pintu dan buzzer sebagai alarm atau sebagai peringatan.

Sistem keamanan berfungsi dengan baik. Sistem keamanan ini dilapis oleh system password dan dapat meng-unlock system keamanan pintu yang lain. Sistem ini tidak dapat bekerja jika secara bersamaan ada seseorang yang ingin membuka pintu maka sistem akan menentukan sistem pintu yang ditekan terlebih dahulu dan sistem pintu tersebut dapat beroperasi dan sistem pintu yang lain menunggu sampai selesai.

Berdasarkan teori dan pembahasan yang telah dijabarkan sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem pengamanan pintu dapat berguna mencegah orang-orang tidak berpeminggiran masuk kedalam ruangan. Sistem keamanan dapat beroperasi jika password yang dimasukkan benar. Sehingga dapat melakukan langkah berikutnya yaitu membuka pintu. Pada saat itu sensor pir akan mendeteksi keberadaan orang di depan pintu. Jika ada seseorang didepan maka pintu akan terbuka secara otomatis dan apabila pintu hampir menutup sedangkan tiba-tiba seseorang berada di depan pintu maka secara otomatis pintu akan terbuka kembali.
2. Dalam sistem ini terdapat aplikasi untuk mengubah password. Tetapi, aplikasi ini tidak ditampilkan karena demi sistem keamanan agar password tidak diganti-ganti oleh yang bukan pemiliknya.
3. Sistem keamanan ini dapat berkomunikasi antara 2 pintu sebab pintu dapat mengremote/ menonaktifkan pintu satunya agar into tersebut tidak dapat beroperasi atau tidak dapat digunakan.

2.3. PROTOTIPE SISTEM KEAMANAN RUMAH MENGGUNAKAN *WEBCAM* DAN *FINGER PRINT* BERBASIS *WEB* DAN *SMS*

Haryadi Amran Darwito (2008), Keadaan rumah tinggal yang nyaman dan aman menjadi idaman setiap keluarga di dunia ini. Kenyamanan dan keamanan yang kita harapkan ini tidak terlepas dari keadaan sosial di lingkungan kita tinggal. Artinya jika lingkungan kita terasa nyaman dan aman untuk kita tinggal secara tidak langsung kenyamanan dan keamanan tersebut akan terasa juga di rumah tinggal kita, oleh karena itu akan dibuatlah suatu sistem keamanan rumah menggunakan webcam dan fingerprint.

Sistem keamanan rumah menggunakan webcam dan finger print ini bekerja apabila limit switch aktif, kemudian memberi trigger ke server untuk mengaktifkan webcam agar pencuri yang memasuki rumah dicapture dan alarm secara otomatis akan aktif. Setiap gambar yang di-capture webcam akan dikirimkan ke web server yang bisa diakses melalui internet. Pada saat alarm aktif server sistem akan mengaktifkan hp (handphone) agar mengirim sms ke pemilik rumah bahwa ada orang lain yang memasuki rumah secara paksa. Untuk mengontrol sistem ini menggunakan finger print sebagai autentikasi setiap anggota keluarga penghuni rumah. Untuk anggota keluarga pemilik rumah yang akan memasuki rumah diharuskan untuk mengakses finger print terlebih dahulu, sehingga apabila sidik jari yang dimasukkan cocok maka hp server mendapat

trigger dari fingerprint dan secara otomatis akan mengirimkan sms pada pemilik rumah apabila yang memasuki rumah adalah anggota pemilik rumah.

berdasarkan studi dan penelitian yang dilakukan pada sebelumnya, maka dapat disimpulkan beberapa hal antara lain:

1. Setiap penghuni rumah harus melakukan registrasi *finger print* agar dapat mengakses sistem keamanan (on/off sistem).
2. Pada program *finger print* digunakan sebagai proses identifikasi user yang akan menggunakan fasilitas ini.
3. Webcam akan aktif apabila *limit switch* terputus / tidak diberi beban.
4. *limit switch* digunakan sebagai trigger untuk mengaktifkan webcam capture, mengirim sms dan menyalakan bel dc (alarm).
5. Yang dapat menyalakan dan mematikan system keamanan rumah melalui sensor finger print hanya penghuni rumah yang sidik jarinya terdaftar.

2.4 INTEGRASI PENGENDALIAN PINTU MASUK RUANG DENGAN SISTEM INFORMASI DATA DENGAN BANTUAN ALAT PEMBACA BARCODE

Harnadi (2005)Makalah ini menawarkan suatu rancangan sistem yang dapat mengintegrasikan pengendalian pintu masuk ruang yang digunakan untuk pembatasan akses masuk dengan sistem informasi data dalam server database. Suatu lembaga yang sebelumnya telah memiliki sistem informasi data yang tersimpan di database dapat menambahkan aplikasi pengendalian pintu masuk

ruang dengan memanfaatkan alat pembaca barcode untuk memasukkan kode akses secara cepat.

Komputer client digunakan untuk menjalankan aplikasi pengendalian pintu masuk dengan kode akses dari alat pembaca barcode dikirimkan ke server database di komputer server untuk dicocokkan. Bila ditemukan, data otoritas tersebut digunakan untuk menggerakkan sistem pintu, dicatat dan disimpan sebagai data pemantauan dan evaluasi.

Pembaca barcode digunakan untuk memasukkan kode akses ke dalam komputer untuk nantinya dicocokkan dengan data pada system informasi data yang dimiliki. Bila kode akses ditemukan maka komputer akan mengeluarkan data pengendalian pintu masuk melalui port I/O yang dimiliki. Integrasi ini dimungkinkan dengan bantuan bahasa pemrograman Delphi yang memiliki kemampuan dalam hal pengolahan database yang handal karena memiliki database engine, kemampuan pengolahan data dan algoritma bahasa aras tinggi dan kemampuan memasukkan dan mengeluarkan data melalui port I/O komputer.

Komputer client bertugas mengeluarkan data pengendalian pintu masuk ruang, mencatat data pengunjung yang melakukan akses ke pintu masuk ruang dan menyimpan data tersebut ke dalam database serta mengolah data tersebut menjadi data pemantauan dan evaluasi.

Dari rancangan yang diulas dimakalah ini dapat disimpulkan:

1. Sistem informasi data dapat diperluas manfaatnya untuk banyak aplikasi seperti pengendali pintu masuk ruang secara otomatis.
2. Pembaca barcode diperlukan untuk memasukkan data kode akses bagi orang yang memiliki otoritas terhadap ruangan.
3. Terdapat banyak metode yang dapat digunakan sebagai pembatas akses masuk ruang seperti dengan menggunakan sidik jari, retina mata, suara, dan wajah. Pemilihan akses masuk dengan barcode untuk kepentingan kepraktisan dan kesederhanaan sistem.
4. Koneksi data diperlukan bagi pencocokkan data kode akses pada server database dikomputer server.
5. Sistem pengendali pintu ruang dapat menggunakan motor stepper maupun motor dc yang masing-masing membutuhkan rangkaian driver/pengendali yang berbeda.
6. Sinyal kendali dari komputer dikeluarkan melalui port paralel LPT dengan menggunakan perintah kendali port yang dibuat menggunakan bahasa pemrograman Delphi.

BAB III

LANDASAN TEORI

3.1 Mengenal Simulasi

Simulasi merupakan salah satu cara untuk memecahkan berbagai persoalan yang dihadapi didunia nyata (*real world*). Banyak metode yang dibangun dalam *operations research* dan *system analyst* untuk kepentingan pengambilan keputusan dengan menggunakan berbagai analisis data.

Linier programming adalah simulasi memberikan hasil yang cukup baik bila digunakan untuk memecahkan berbagai persoalan, termasuk dalam pembuatan perencanaan kegiatan. Namun demikian apabila untuk waktu yang panjang lebih tepatnya bila menggunakan multi periode linier programming yang didasarkan pada perhitungan tahunan dan kemudian untuk peninjauan bulanan dan yang diambil dari rata-rata tahunan. Dengan demikian tidak akan di lakukan peninjauan atas hari-hari perencanaan tersebut, termasuk apabila perencanaan ini untuk jangka waktu pendek.

Terdapat persoalan lain didalam linier programming, yaitu adanya ketidak pastian, yang diartikan bahwa tidak banyak berperan lagi karena dianggap tidak memadai dali untuk digunakan untuk menentukan dederapa variable didalam modalnya sendiri. Kekurangan tersebt dapat diatasi dengan *sensitivityanalysis*, walapun perhitungannya tidak banyak menolong, Simulasi dapat diartikan sebagai suatu system yang digunakan untuk memecahhkan atau menguraikan persoalan-persoalan dalam kehidupan

nyata yang penuh dengan ketidak pastian dengan tidak atau menggunakan model atau metode tertentu dan lebih ditekankan pada pemakaian computer untuk mendapat solusinya.

Pada pendekatan simulasinya, agar dapat menyelesaikan persoalan yang rumit akan lebih mudah dilakukan bila dimulai dengan membangun model percobaan dari suatu system. Untuk melakukannya kita perlu memperhatikan tiga unsure penting didalam pemodelan simulasi, yaitu *system, entitas dan attributes*.

Adapun keuntungan didalam penggunaan simlasi

1. menghemat waktu
2. dapat melebar laskan waktu
3. dapat mengatasi sumber yang bervariasi
4. Mengoreksi kesalahan-kesalahan perhitungan
5. Dapat dihentikan dan dijalankan kembali
6. Mudah diperbanyak

(Kakiay, 2003)

3.2 Borland Delphi

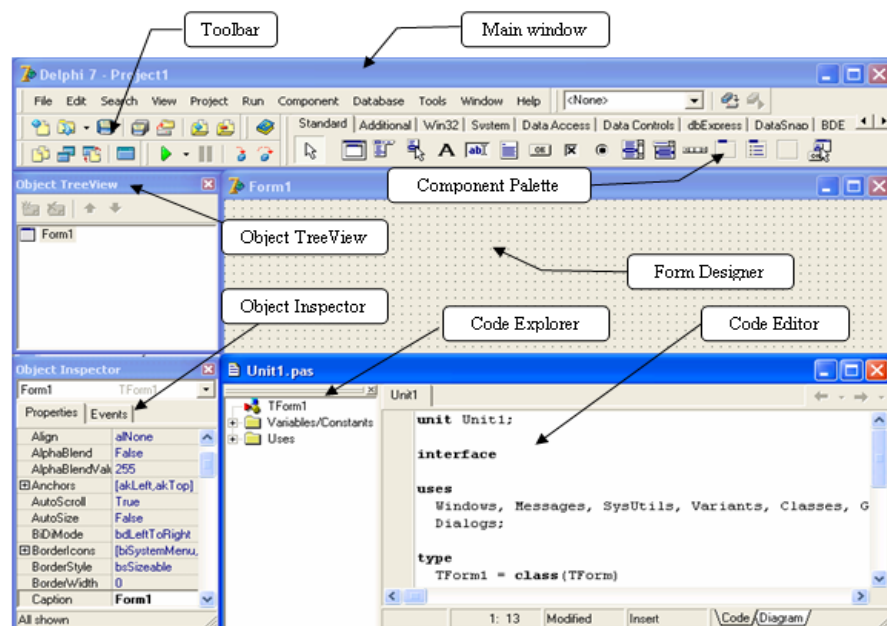
Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yan menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari

Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows.

Khusus untuk pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format database yang dimiliki Delphi adalah format database Paradox, dBase, MS Access, ODBC, SyBASE, Oracle dan lain-lain.

3.2.1. Mengenal IDE Delphi

IDE Delphi adalah Lingkungan pengembangan terpadu atau Integrated Development Environment (IDE) dalam program Delphi terbagi menjadi delapan bagian utama, yaitu Main Window, ToolBar, Component Palette, Form Designer, Code Editor, Object Inspector, Exploring, dan Object TreeView. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini :



Gambar 3.1. Lembar Kerja Borland Delphi

(Andi; 2002 : 5)

a. Main Window

Jendela utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, berupa Main menu, Toolbar dan Component palette.

(Andi; 2002 :5)

b. Main Menu

Menu utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi Windows lainnya. Dengan menggunakan fasilitas menu dapat memanggil atau menyimpan program. Pada dasarnya perintah yang diberikan dapat ditemukan pada bagian menu utama ini.

(Andi; 2002 : 6)

c. Toolbar

Delphi memiliki beberapa toolbar yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian toolbar berfungsi sebagai pengganti suatu menu perintah yang sering digunakan. Toolbar sering disebut juga dengan speedbar.

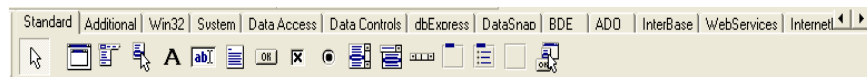
Toolbar terletak pada bagian bawah baris menu. Pada kondisi default delphi memiliki enam bagian toolbar, antara lain: standart, view, debug, desktops, custom dan component palette.

(Andi; 2002 :6)

d. Component Palette

Component Palette berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada VCL (Visual Component Library).

(Andi; 2002 : 7)

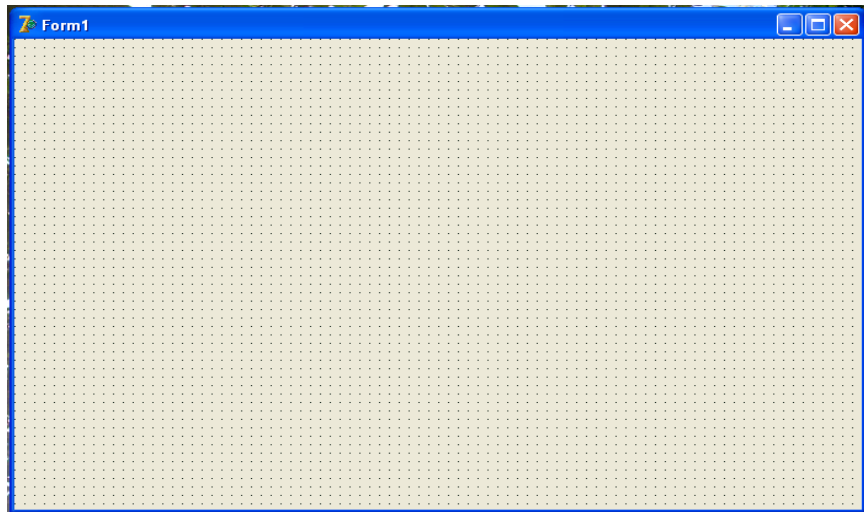


Gambar 3.2. *Component Palette*

e. Form Designer

Form Designer merupakan suatu object yang dapat dipakai sebagai tempat untuk merancang program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari component palette. Dalam form terdapat titik-titik yang disebut grid yang berguna untuk membantu pengaturan tata letak object yang dimasukkan dalam form. Sebuah form mengandung unit yang berfungsi untuk mengendalikan form dan dapat juga mengendalikan komponen-komponen yang terletak dalam form dengan menggunakan Object Inspector dan Code Editor.

(Andi; 2002: 8)

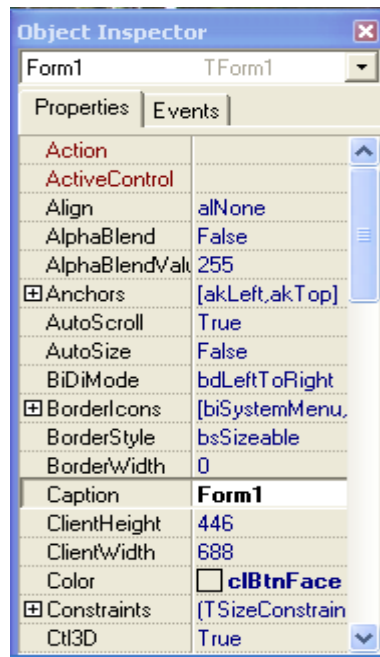


Gambar 3.3. *Form Designer*

f. Object Inspector

Object Inspector digunakan untuk mengubah propertis atau karakteristik dari sebuah komponen. Object Inspector terdiri dari dua tab yaitu Properties yang berfungsi mengubah sifat – sifat yang dibawa oleh object tersebut sedangkan Events berfungsi sebagai daftar procedure yang dilakukan oleh object tersebut ketika suatu peristiwa terjadi.

(Andi ; 2002 : 9)

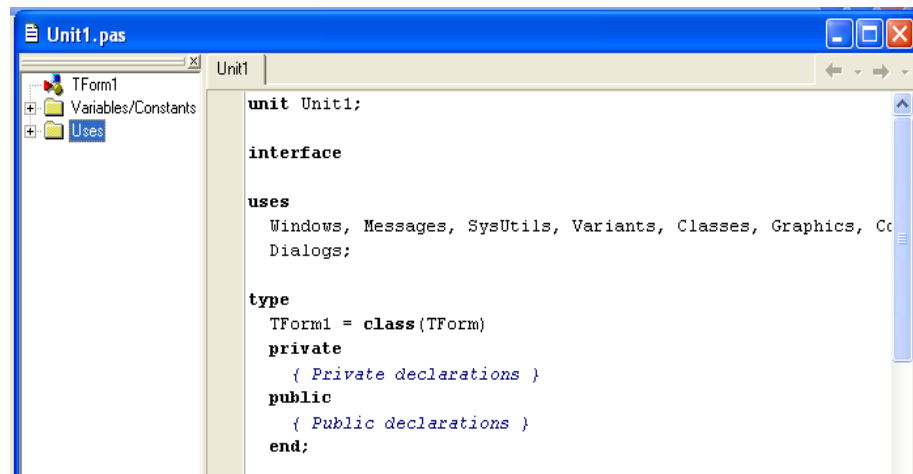


Gambar 3.4. *Object Inspector*

g. *Code Editor*

Code Editor merupakan suatu tempat di mana dapat menuliskan kode program. Satu program mungkin mempunyai satu atau lebih unit. Dalam Borland Delphi, ada satu unit yang tak terpisahkan dengan form dan setiap kali dibuat satu form maka otomatis pula dibuat satu unit. Unit yang berhubungan dengan form ini, biasanya dipakai untuk mengatur dan mengendalikan segala sesuatu yang berhubungan dengan form. Selain unit jenis ini, dapat pula dibuat unit yang sama sekali terpisah dengan form. Unit jenis ini dapat berisi kumpulan fungsi atau procedure yang dipakai program aplikasi.

(Andi; 2002 : 11)

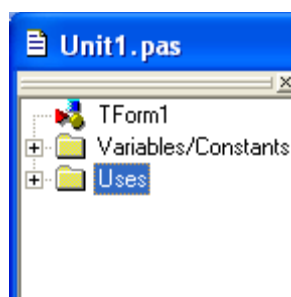


Gambar 3.5. Code Editor

h. Code Explorer

Code Explorer digunakan untuk memudahkan pemakai berpindah antar file unit yang terdapat di dalam jendela Code Editor. Jendela Code Explorer berisi diagram pohon yang menampilkan semua tipe, class, properti, method, variabel global, dan rutin global yang telah didefinisikan di dalam unit.

(Andi; 2002 :12)

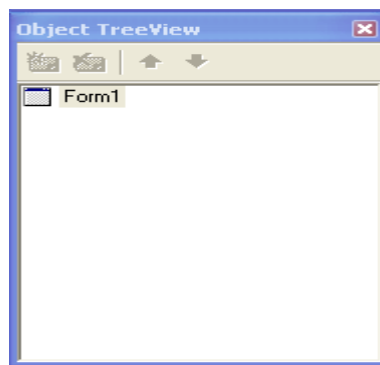


Gambar 3.6. Code Explorer

i. *Object TreeView*

Object TreeView menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. Object TreeView juga menampilkan hubungan logika antar komponen.

(Andi; 2002 :12)



Gambar 3.7. *Object TreeView*

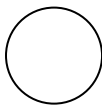

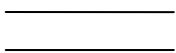
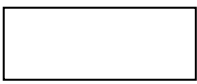
3.3 Analisa System

3.3.1. Data Flow Diagram (DFD)

Data Flow Diagram merupakan alat pemodelan data yang menggambarkan sistem sebagai suatu jaringan dari fungsi-fungsi atau proses-proses dari sistem yang saling berhubungan satu sama lain dengan aliran data yang digambarkan dengan anak panah. Simbol-simbol dalam data flow diagram ditunjukkan pada tabel 2.1. yaitu :

Tabel 2.1. Simbol Data Flow Diagram

(Jogiyanto, 2002)

No.	Simbol	Keterangan
1.	Proses 	Simbol ini digunakan untuk proses pengolahan atau tranformasi data
2.	Aliran Data 	Menggambarkan perpindahan berupa data, atau paket informasi dari satu bagian sistem ke bagian lain
3.	Data Store (Simpanan Data) 	Menggambarkan model dari kumpulan paket data yang tersimpan
4.	Terminator (Eksternal Entity) 	Menggambarkan kesatuan luar yang berhubungan dengan sistem (Menggambarkan asal data atau tujuan)

3.3.2. Entity Relationship Diagram

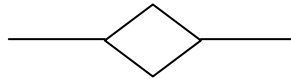
1. Entitas adalah orang, tempat kejadian dan konsep datanya dapat terekam. Dalam hal ini sebagai contoh adalah entitas untuk sopir yang memuat banyak atribut dan data value. Entitas tersebut digambarkan dengan empat persegi panjang.



Gambar 2.5. Simbol Entitas

(Jogiyanto, 2002)

2. Relasi adalah hubungan antara dua entitas atau lebih. Simbol relasi ditunjukkan seperti pada gambar 2.6.



Gambar 2.6. Simbol Relasi

(Jogiyanto, 2002)

Dalam penggambaran Entitas Relationship Diagram ditentukan oleh tiga faktor :

1. Derajat (*Degree*)

Derajat digunakan untuk mengenalkan banyak entitas yang terlibat dalam entitas. Terdiri dari :

- a. *Unary*

Suatu entitas yang memiliki relasi dengan dirinya sendiri.

- b. *Binary*

Relasi antar dua entitas yang berbeda.

- c. *Ternary*

Suatu relasi antara tiga entitas yang berbeda

2. Hubungan (*Connectivity*)

Hubungan adalah setiap kumpulan relasi yang berpasangan antara himpunan entitas yang satu dengan yang lain. Relasi kedua entitas tersebut salah satu dari relasi berikut :

a. Satu Ke Satu (*One To One*)

Yaitu hubungan antara entitas yang pertama dan kedua merupakan satu berbanding satu.

b. Satu Ke Banyak (*One To Many*)

Yaitu hubungan antara entitas yang pertama dan kedua merupakan satu berbanding banyak atau sebaliknya.

c. Banyak Ke Banyak (*Many To Many*)

Yaitu hubungan antara entitas pertama dan kedua adalah banyak berbanding banyak.

(Jogiyanto, 2002)

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisa system

Pada saat ini sistem penggunaan ruang kelas yang sedang berjalan dilakukan dengan cara manual yaitu hanya membuka dan menutup pintu ruangan kelas. Mahasiswa dan dosen bisa menggunakan ruangan dengan melihat jadwal terlebih dahulu, karena jadwal adalah pedoman untuk penggunaan ruangan.

4.1.1. Narasi Proses Alir Dokumen Pada Sistem Lama

1. Mahasiswa dan dosen harus terdaftar/aktif didalam tahun ajaran yang sedang berjalan.
2. Mahasiswa harus mengambil matakuliah, karena diperlukan dalam pembuatan jadwal.
3. Mahasiswa dan dosen menggunakan ruangan sesuai dengan jadwal belajar mengajar yang sudah ditentukan,.
4. Pengguna dapat memasuki ruangan yang sudah tidak terkunci dengan cara manual, yaitu hanya membuka dan menutup pintu saja.
5. Mahasiswa dan dosen harus mengisi laporan masuk ruangan yang telah tersedia dengan cara manual sebagai bukti bahwa ruangan itu benar-benar sudah digunakan sesuai jadwal.
6. Hasil laporan masuk ruangan diberikan BAAK agar dapat membantu dalam pengolahan data.

4.1.2. Kelebihan dan Kekurangan Pada Sistem Lama

1. Kelebihan-Kelebihan Pada Sistem Lama

- a) Tidak memerlukan biaya yang banyak dalam perancangan penggunaan ruangan di Unisbank.
- b) Pengguna dapat menggunakan ruangan kosong dengan bebas.
- c) Penggunaan ruangan dilakukan dengan cepat dalam penyampaian hasil laporan penggunaan ruangan.

2. Kekurangan-Kekurangan Pada Sistem Lama

- a) Sering terjadinya tindak kriminal didalam ruangan.
- b) Masih menggunakan sistem manual dalam penggunaan ruangan.
- c) Kurangnya disiplin waktu didalam penggunaan ruangan.
- d) Pengguna dapat memanipulasi laporan kehadiran ruangan.

4.1.3. Identifikasi Masalah

Melihat dari sistem yang sedang berjalan di Fakultas Teknik Informatika di Universitas Stikubank (Unisbank) Semarang ini masih perlu dikembangkan. Hal ini disebabkan oleh beberapa faktor seperti :

1. Penggunaan ruangan masih menggunakan sistem manual untuk membuka dan menutup pintu ruangan kelas.
2. Sering kali terjadinya manipulasi didalam monitoring kehadiran ruangan.
3. Sering terjadi kesalahan didalam memasuki ruangan.

4.1.4. Identifikasi Kebutuhan Data Dan Informasi

1. Identifikasi Data

- a) Data Matakuliah
- b) Data Ruangan
- c) Data Dosen
- d) Data Mahasiswa

2. Identifikasi Informasi

- a) Informasi jadwal
- b) Informasi kehadiran masuk ruangan

4.1.5. Alternatif Sistem Yang Diusulkan

Dari hasil analisa sistem penggunaan ruangan yang sedang berjalan pada saat ini penulis akan melakukan pengembangan sistem penggunaan ruangan berbasis komputerisasi yang menggunakan sensor sidik jari sebagai alat untuk mengakses masuk ruangan sehingga dapat membantu didalam pengontrolan penggunaan ruangan yang dapat dilakukan oleh BAAK.

4.1.6. Identifikasi Kebutuhan Hardware

Adapun spesifikasi standar minimal perangkat keras yang akan digunakan didalam simulasi sistem tersebut adalah sebagai berikut :

1. Personal Computer (PC)

Teknologi PC dapat dipertimbangkan menjadi suatu sistem dapat bekerja yang terpadu, dengan demikian pengolahan data dapat disentralisasikan dan akan menciptakan efisiensi kerja dan

validasi data serta informasi akan selalu tersedia dan aman. Adapun

rincian PC yang dipakai adalah :

- a) Minimal Pentium IV 3,06 GHz
- b) Memory 256 MB
- c) Hardisk dengan kapasitas 80 GB
- d) Disk drive
- e) CD RW 52X
- f) Mouse dan keyboard standar

2. Printer

Printer yang digunakan sebaiknya printer yang mempunyai kecepatan cetak yang tinggi dan tidak terlalu mahal.

3. UPS (*Uninterruptible Power Supply*)

Pemakaian UPS berguna untuk menghindari dari putusnya aliran listrik sehingga kehilangan data dapat dihindari.

4. Simulasi Alat

- a) Ruang Beserta Pintu Ruang

Tempat dimana simulasi sistem penggunaan ruangan dilakukan dan akan dijalankan berdasarkan sistem yang akan dibuat.

- b) Microcontroller

Mikrokontroler yang sudah diprogram yang berfungsi untuk mengendalikan alat untuk membuka pintu secara otomatis pada ruangan. Mikrokontroler berfungsi sebagai menerima input dari

computer server didalam memproses simulasi kemudian memberikan reaksi ke output.

c) Catu Daya

Perancangan catu daya ini menggunakan trafo 2A untuk menghasilkan keluaran sebesar 12V dc dan 5V dc dengan arus keluaran. Keluaran 12V dc digunakan untuk mengaktifkan motor ,sedangkan 5V digunakan untuk mengaktifkan mikrokontroler.

d) Buzzer

Buzzer merupakan speaker atau device yang digunakan untuk mengeluarkan suara atau bunyi, bunyi yang dihasilkan ini hanya satu nada. Buzzer pada alat ini berfungsi untuk mengeluarkan suara yang didapat dari menginisialisasi masukan dari fingerprint yang menandakan bahwa pintu pada ruangan sudah bisa dibuka.

e) Fingerprint

Fingerprint (sensor sidik jari) adalah sensor mesin yang berfungsi untuk membaca bentuk sidik jari dan kemudian data yang sudah di encrypt akan dikirim melalui kabel USB menuju terminal untuk pengolahan data.

4.1.7. Identifikasi Kebutuhan Perangkat Lunak

Untuk dapat beroperasi secara optimal, komputer tersebut harus memenuhi spesifikasi yang diperlukan serta diperlukan dukungan dari

perangkat lunak (*software*) sesuai dengan kebutuhannya agar program aplikasi dapat berjalan dengan baik.

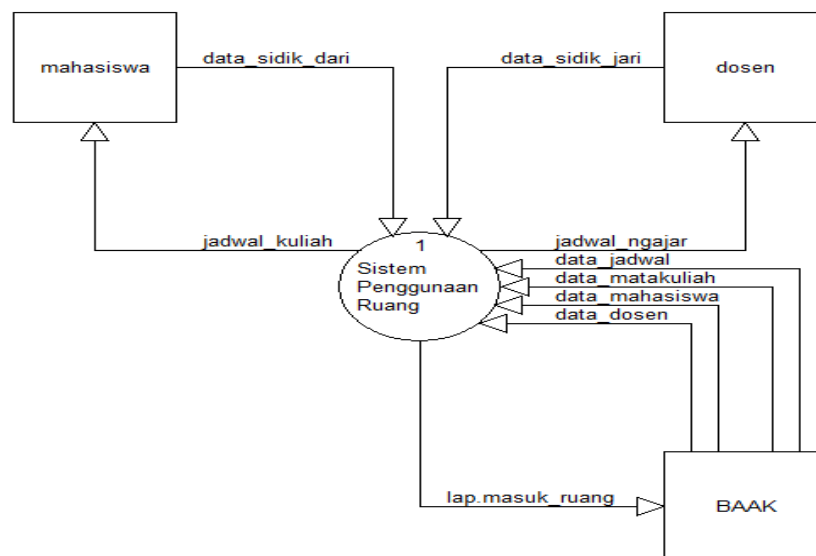
Kebutuhan perangkat lunak tersebut antara lain :

- a. Microsoft Windows XP yang digunakan sistem operasi.
- b. Bahasa pemrograman Borland Delphi 6

4.2. Perancangan Sistem

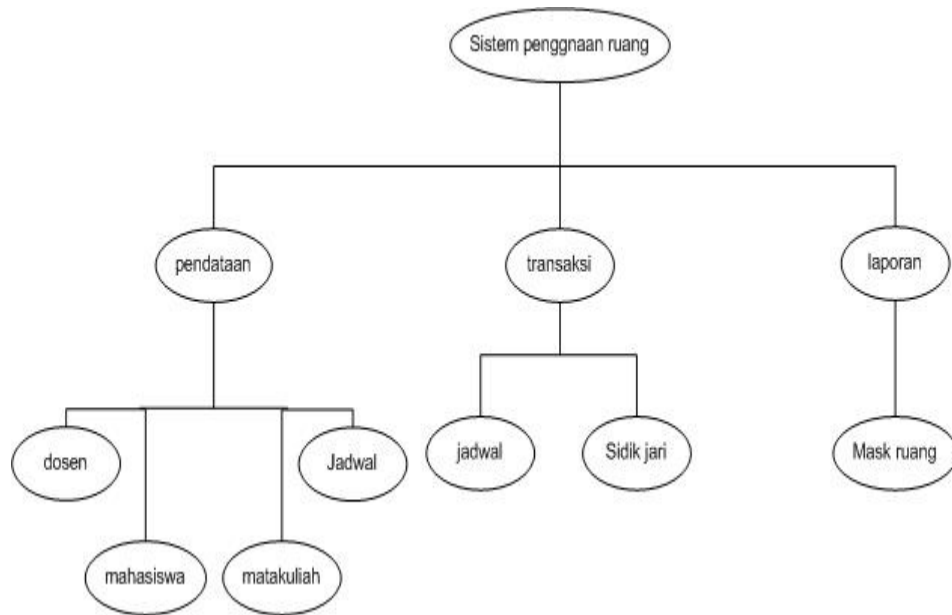
Setelah melakukan beberapa pengamatan terhadap sistem yang sedang berjalan, maka sistem penggunaan ruang secara manual yang dijalankan sudah tidak efisien lagi dari segi waktu dan tidak efektif lagi dari segi pengolahan datanya. Untuk memecahkan persoalan tersebut, maka dibutuhkan pembaharuan dari sistem manual yang sedang berjalan dengan diterapkan menjadi sistem yang lebih efektif dan efisien, sehingga proses yang tadinya memakan waktu yang cukup lama dapat dikerjakan lebih cepat dan akurat.

4.2.1. Context Diagram



Gambar 4.1. Context Diagram Penggunaan Ruang

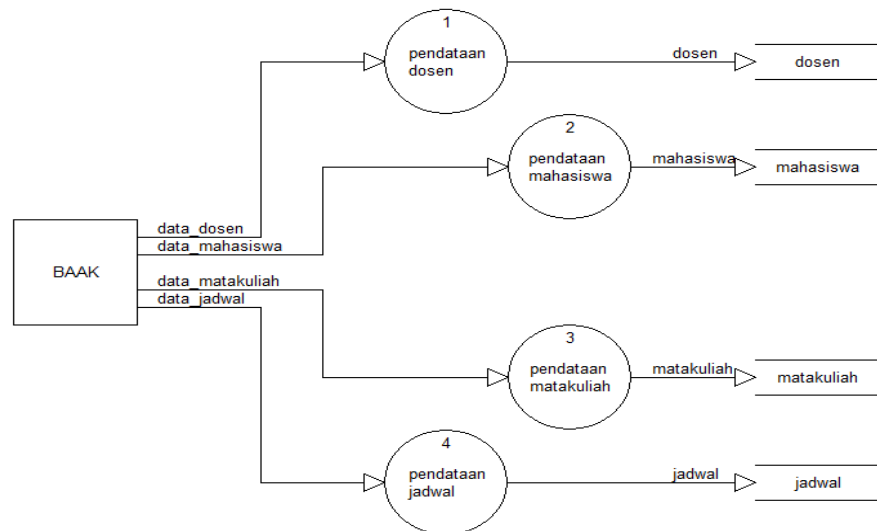
4.2.2. Diagram Decomposisi



Gambar 4.2. Diagram Decomposisi

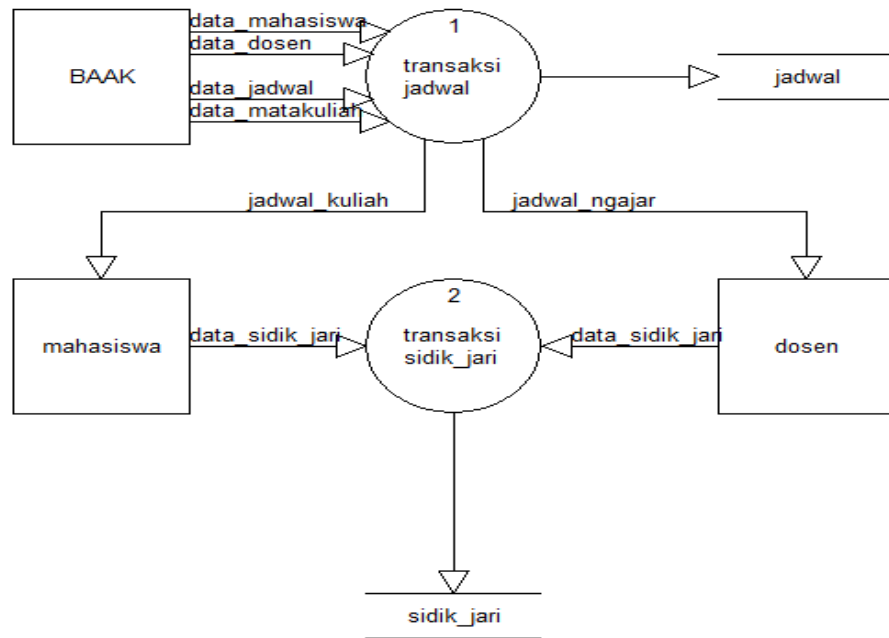
4.2.3. DFD levelled

a. DFD level 1 pendataan



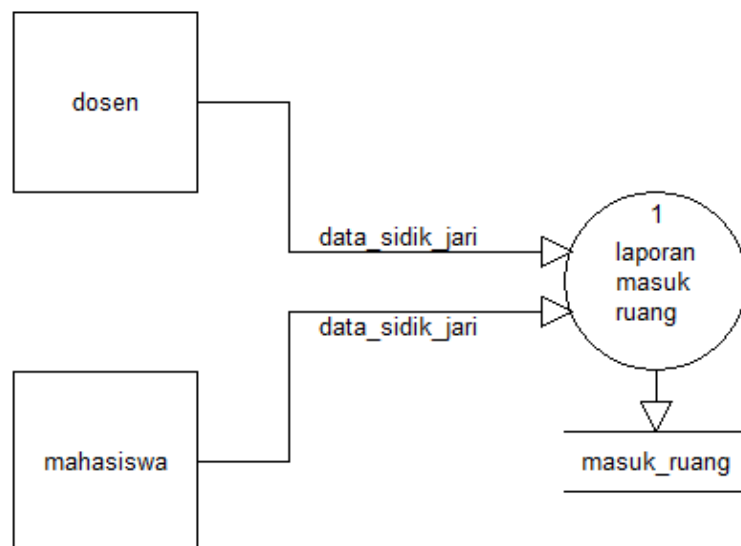
Gambar 4.3 DFD Level 1 Pendataan

b. DFD level 2 transaksi



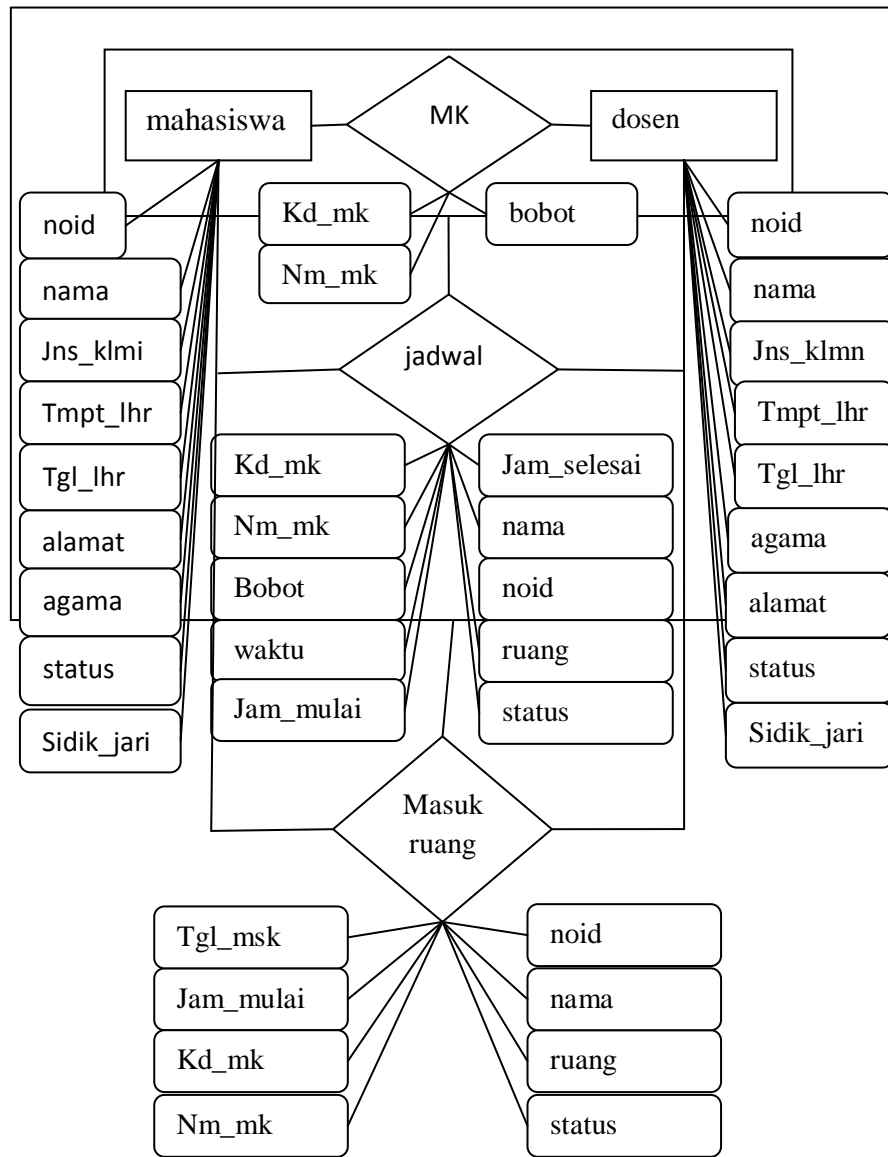
Gambar 4.4. DFD level 2 transaksi

c. DFD level 3 laporan



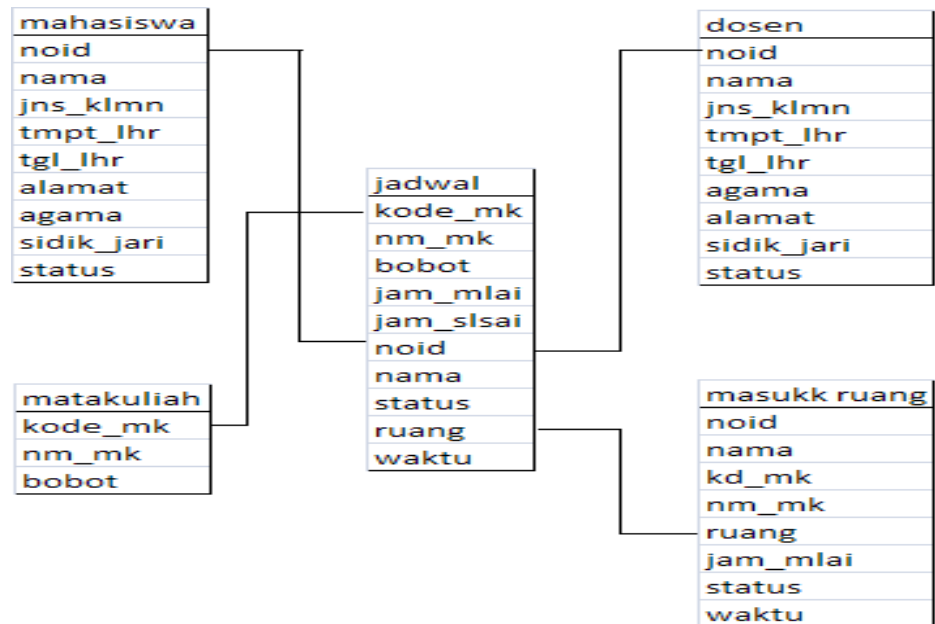
Gambar 4.5. DFD level 3 laporan

4.2.4. ERD (Entity Relationship Diagram)



Gambar 4.6. Entity Relationship Diagram

4.2.5. Relationship Tabel



Gambar 4.7 Relationship Tabel

1. Data Dictionary

a. Tabel Matakuliah

Tabel 4.1 : Tabel Matakuliah

No	Nama field	Size	Tipe	Keterangan
1	Kd_mk	15	Varchar	Kode matakuliah
2	Nm_mk	35	Varchar	Nama matakuliah
3	Bobot (sks)	20	Varchar	Jumlah sks

b. Tabel Mahasiswa

Tabel 4.2. Tabel Mahasiswa

No	Nama field	Siza	Type	Keterangan
1	Noid	15	Varchar	No induk user
2	Nama	35	Varchar	Nama
3	Sidik_jari		Longblab	Sensor sidik jari
4	Jns_klmn	20	Varchar	Jenis kelamin
5	Tmpt_lhr	20	Varchar	Tempat lahir
6	Tgl_lhr		Date	Tanggal lahir
7	Agama	25	Varchar	Agama
8	Alamat	45	Varchar	Alamat
9	Status	20	Varchar	Status

c. Tabel Dosen

Tabel 4.3. Tabel Dosen

No	Nama	Size	Type	Keterangan
1	Noid	15	Varchar	No induk user
2	Nama	35	Varchar	Nama dosen
3	Jns_klmn	20	Varchar	Jenis kelamin
4	Tmpt_lhr	20	Varchar	Tempat lahir
5	Tgl_lhr		Date	Tanggal lahir
6	Agama	25	Varchar	Agama

7	Alamat	45	Varchar	Alamat
8	Status	20	Varchar	Status
9	Sidik_jari		Longblob	Data sidik jari

d. Tabel Jadwal Kuliah

Tabel 4.4 : Tabel Jadwal Kuliah

No	Nama field	Size	Tipe	Keterangan
1	Kd_mk	15	Varchar	Kode matakuliah
2	Nm_mk	35	Varchar	Nama matakuliah
3	Bobot	20	Varchar	Jumlah sks
4	Waktu		Date	Waktu
5	Jam_mulai		Time	Jam mulai
6	Jam_selesai		Time	Jam selesai
7	Nama	35	Varchar	Nama user
8	Noid	15	Varchar	No induk user
9	Ruang	25	Varchar	Ruangan
10	Status	20	Varchar	Status

e. Tabel Masuk Ruang

Tabel 4.5 : Tabel Masuk Ruang

No	Nama field	Size	Tipe	Keterangan
1	Tgl_masuk		Date	Tanggal masuk
2	Jam_mulai		Time	Jam masuk
3	Kd_mk	15	Varchar	Kode matakuliah
4	Nm_mk	25	Varchar	Nama matakuliah
5	Noid	15	Varchar	No induk user
6	Nama	35	Varchar	Nama user
7	Ruang	25	Varchar	Ruang
8	Status	15	Varchar	Status

4.2.6. Data Dictionary (Kamus Data)

1. Tabel Mahasiswa

Mahasiswa = @noid + nama + sidik_jari + jns_klmn + tmp_lhr
+ tgl_lhr+ alamat + agama + status.

noid = 15 Varchar 15

nama = 1 Varchar 35

sidi_jari = Longblob

jns_klmn = 1 Varchar 20

tmp_lhr = 1 Varchar 20

tgl_lhr = Date

alamat = 1 Varchar 45

agama = 1 Varchar 25

status = 1 Varchar 20

2. Tabel Dosen

Dosen = @noid+ nama + jns_klmn + tmpt_lhr + tgl_lhr +
agama + alamat + status + sidik_jari.

noid = 15 Varchar 15

nama = 1 Varchar 35

jns_klmn = 1 Varchar 20

tmpt_lhr = 1 Varchar 20

tgl_lhr = date

agama = 1 Varchar 25

alamat = 1 Varchar 45

status = 1 Varchar 20

Sidik_jari = Longblob.

3. Tabel Jadwal kuliah

Jadwal kuliah = @ kd_mk + nm_mk + bobot + waktu +
jam_mulai + jam_selesai + nama + noid +
ruang + status.

Kd_mk = 1 Varchar 15

Nm_mk = 1 Varchar 35

bobot = 1 Varchar 20

waktu = date
jam_mulai = time
jam_selesai = time
nama = 1 Varchar 35
noid = 1 Varchar 15
ruang = 1 Varchar 25
status = 1 Varchar 20

4. Tabel matakuliah

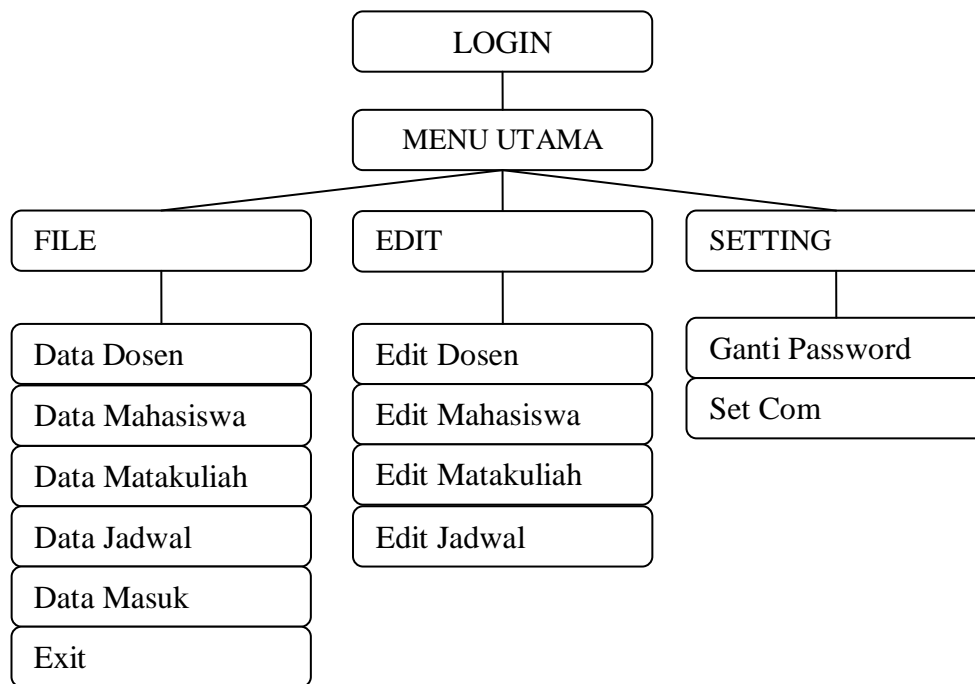
matakuliah = @kd_mk + nm_mk + bobot.
Kd_mk = 1 Varchar 15
nm_mk = 1 Varchar 35
bobot = 1 Varchar 20

5. Tabel Masuk Ruang

Masukruang = @ tgl_masuk + jam_mulai+ kd_mk + nm_mk
+ noid + nama+ ruang + status.
Tgl_masuk = date
Jam_mulai = time
Kd_mk = 1 Varchar 15
Nm_mk = 1 Varchar 35
noid = 1 Varchar 15
nama = 1 Varchar 35
ruang = 1 Varchar 25
status = 1 Varchar 15

4.3. Perancangan Tampilan

Perancangan tampilan merupakan suatu kesatuan kerangka prosedur dari program itu sendiri, dimana tampilan dalam suatu sistem perangkat lunak akan saling berhubungan dan membentuk suatu jaringan tampilan antar menu. Jaringan tampilan antar menu merupakan suatu diagram yang menggambarkan hubungan atau jalur kerja tampilan kerangka aliran prosedur sistem.



Gambar 4.8 Struktur Menu Utama

Struktur menu utama terdiri dari login admin yang terdiri dari menu file yang terdiri dari sub menu log out, menu edit yang terdiri dari sub menu mahasiswa, dosen, matakuliah, jadwal kuliah, menu laporan terdiri dari sub menu penggunaan ruangan.

4.3.1. Perancangan Input Output

Perangkat input output yang membangun perangkat lunak sebagai tindak lanjut dari implementasi program yang di bangun, terdapat beberapa input output yang akan dibuat dalam system ini.

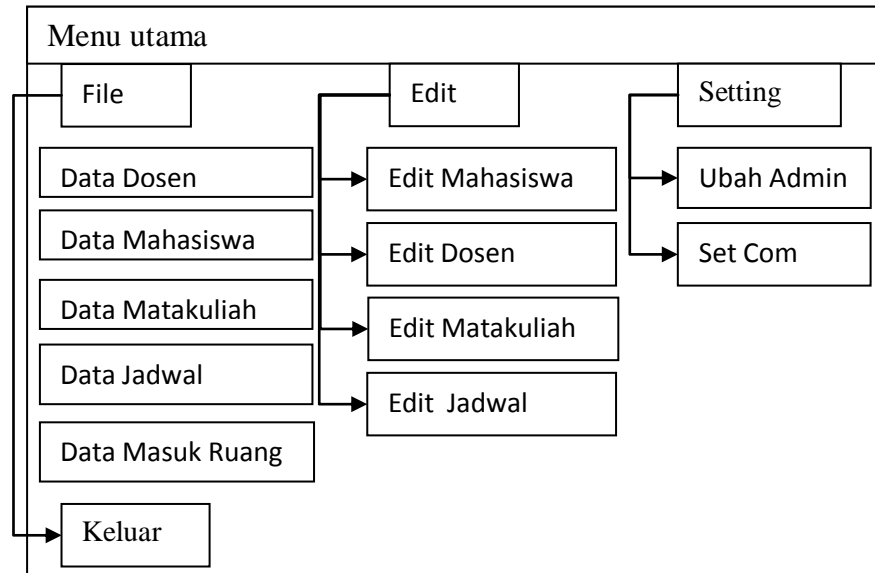
1. Tampilan form login

Form Login	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="OK"/>	<input type="button" value="Batal"/>

Gambar 4.9 Desain Login

Pada form login akan memberi inputan ID user dan password dan terdapat tombol OK agar nantinya dapat digunakan untuk membuka menu utama dan menjalankan program.

2. Perancangan Menu Utama



Gambar 4.10 Desain Menu Utama

3. Perancangan Form Data Dosen

Form Data Dosen					
Cari Data					
Nama	<input type="text"/>	Alamat	<input type="text"/>		
Nik	nama	alamat	Jns kelamin	agama	status
<input type="text"/>					
<input type="button" value="Keluar"/>					

Gambar 4.11 Desain Tampilan Form Dosen

Pada form data dosen digunakan untuk menampilkan data dosen yang sudah tersimpan dan pencarian berdasarkan nama

dosen dan alamat. Pada form ini terdapat tombol keluar yang digunakan untuk keluar dari form dosen atau kembali ke menu awal.

5. Perancangan Form data mahasiswa

form data mahasiswa					
Cari data					
nama	<input type="text"/>	alamat	<input type="text"/>		
nim	nama	alamat	Jenis kelamin	agama	status
<input type="text"/>					
					Keluar

Gambar 4.12 Desain Tampilan Form Mahasiswa

Pada form data Mahasiswa digunakan untuk menampilkan data mahasiswa yang sudah tersimpan dan pencarian berdasarkan nama mahasiswa dan alamat. Pada form ini terdapat tombol keluar yang digunakan untuk keluar dari form mahasiswa atau kembali ke menu awal.

6. Perancangan Form Matakuliah

Form Inputan Data Matakuliah		
Cari data		
Nama MK	<input type="text"/>	
Kode MK	Nama MK	SKS
<input type="text"/>		
		<input type="button" value="Keluar"/>

Gambar 4.13 Desain Tampilan Form Matakuliah

Pada form data matakuliah digunakan untuk menampilkan data matakuliah yang sudah tersimpan dan menampilkan pencarian berdasarkan nama matakuliah. Pada form ini terdapat tombol keluar yang digunakan untuk keluar dari form matakuliah atau kembali ke menu awal.

7. Perancangan Form Jadwal

Lembar Data Jadwal					
Cari data					
Nama dos	<input type="text"/>	Nama mhs	<input type="text"/>		
Kd_mk	Nama_mk	dosen	mhs	waktu	ruang
					<input type="button" value="Keluar"/>

Gambar 4.14 Desain Tampilan data Jadwal

Pada form data jadwal digunakan untuk menampilkan data jadwal yang sudah tersimpan dan menampilkan pencarian berdasarkan nama dosen dan nama mahasiswa. Pada form ini terdapat tombol keluar yang digunakan untuk keluar dari form jadwal atau kembali ke menu awal.

8. Perancangan Form data Masuk Ruang

Form Data Masuk Ruangan						
Cari data						
nama	<input type="text"/>	status	<input type="text"/>			
waktu	MK	nim	nik	nama	ruang	status
<input type="text"/>						
						Keluar

Gambar 4.15 Desain Tampilan data masuk ruang

Perancangan form laporan masuk ruang digunakan untuk menampilkan siapa saja yang masuk ruangan disaat jadwal penggunaan ruangan berlangsung, pada form laporan masuk ruang terdapat tombol Keluar yang digunakan untuk keluar dari form masuk ruang atau kembali ke Form Menu Utama.

9. Perancangan Form Edit Dosen

No Induk	<input type="text"/>					
Nama	<input type="text"/>	Registrasi sidik jari				
foto	<input type="text"/>	<input type="text"/>				
Tempat Dan Tgl Lahir	<input type="text"/>	<input type="text"/>				
Agama	<input type="text"/>					
Jenis kelamin	<input type="text"/>					
alamat	<input type="text"/>					
Status	<input type="text"/>					
Nim	Nama	Jenis kelamin	Tmpt lhr	Tgl lhr	Agama	Alamat
<input type="text"/>						
Lalu	Hapus	Hapus	Simpan			
Berikut	Tambah	Batal				Keluar

Gambar 4.16 Desain Tampilan Edit Dosen

Perancangan form Edit Dosen digunakan untuk menginput data dosen pengajar, pada form edit dosen terdapat 8 tombol yaitu tombol Tambah yang digunakan untuk menambah data dosen, tombol Ubah digunakan untuk mengubah data Dosen yang sudah tersimpan, tombol Hapus digunakan untuk menghapus data dosen

yang telah disimpan, tombol Batal digunakan untuk membatalkan pengisian data dosen, tombol simpan untuk menyimpan data dosen, tombol Lalu digunakan untuk memindah cursor sebelumnya, tombol berikut digunakan untuk memindah cursor sesudahnya dan tombol Keluar digunakan untuk keluar dari form dosen

10. Perancangan Form Edit Mahasiswa

No Induk		Registrasi sidik jari														
Nama																
foto																
Tempat Dan Tgl																
Agama																
Jenis kelamin																
alamat																
Status																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 12.5%;">Nim</th> <th style="width: 12.5%;">Nama</th> <th style="width: 12.5%;">Jns kelamin</th> <th style="width: 12.5%;">Tmpt</th> <th style="width: 12.5%;">Tgl</th> <th style="width: 12.5%;">Agama</th> <th style="width: 12.5%;">Alamat</th> </tr> </thead> <tbody> <tr> <td colspan="7" style="height: 40px;"></td> </tr> </tbody> </table>			Nim	Nama	Jns kelamin	Tmpt	Tgl	Agama	Alamat							
Nim	Nama	Jns kelamin	Tmpt	Tgl	Agama	Alamat										
Lalu	Ubah	Hapus	Simpan													
Berikut	Tambah	Batal			Keluar											

Gambar 4.17 Desain Tampilan Edit Mahasiswa

Perancangan form Edit mahasiswa digunakan untuk tampilkan input data mahasiswa, pada form edit mahasiswa terdapat 8 tombol yaitu tombol Tambah yang digunakan untuk menambah data mahasiswa, tombol Ubah digunakan untuk mengubah data mahasiswa yang sudah tersimpan, tombol Hapus digunakan untuk menghapus data mahasiswa yang telah disimpan, tombol Batal digunakan untuk membatalkan pengisian data mahasiswa, tombol simpan untuk menyimpan data mahasiswa, tombol Lalu digunakan untuk memindah cursor sebelumnya, tombol berikut digunakan untuk memindah cursor sesudahnya dan tombol Keluar digunakan untuk keluar dari form mahasiswa.

11. Perancangan Form Edit Matakuliah

Lembar Edit Matakuliah			
Kode Matakuliah		<input type="text"/>	
Nama Matakuliah		<input type="text"/>	
SKS		<input type="text"/>	
Kode MK	Nama MK	SKS	
<input type="text"/>			
Lalu	Ubah	Batal	
Beriku	Tambah	Simpan	Keluar

Gambar 4.18 Desain Tampilan Edit Matakuliah

Perancangan form Edit matakuliah digunakan untuk tampilkan input data matakuliah, pada form edit matakuliah terdapat 8 tombol yaitu tombol Tambah yang digunakan untuk menambah data matakuliah, tombol Ubah digunakan untuk mengubah data matakuliah yang sudah tersimpan, tombol Hapus digunakan untuk menghapus data matakuliah yang telah disimpan, tombol Batal digunakan untuk membatalkan pengisian data matakuliah, tombol simpan untuk menyimpan data matakuliah, tombol Lalu digunakan untuk memindah cursor sebelumnya, tombol berikut digunakan untuk memindah cursor sesudahnya dan tombol Keluar digunakan untuk keluar dari form matakuliah.

12. Perancangan Form Edit Jadwal

Tgl kuliah					
Jam masuk kuliah					
jam selesai kuliah					
Kode matakulai					
matakuliah					
ID dosen					
Nama Dosen					
ID Mhs					
Nama Mhs					
Ruang					
TGL	Jam masuk	Jam selesai	Kode MK	MK	ID Dosen
Lalu	Ubah	Hapus	Simpan		
Berikut	Tambah	Batal	Keluar		

Gambar 4.19 Desain Tampilan Edit Jadwal

Perancangan form Edit jadwal digunakan untuk tampilkan input data jadwal, pada form edit jadwal terdapat 8 tombol yaitu tombol Tambah yang digunakan untuk menambah data jadwal, tombol Ubah digunakan untuk mengubah data jadwal yang sudah tersimpan, tombol Hapus digunakan untuk menghapus data jadwal yang telah disimpan, tombol Batal digunakan untuk membatalkan

pengisian data jadwal, tombol simpan untuk menyimpan data jadwal, tombol Lalu digunakan untuk memindah cursor sebelumnya, tombol berikut digunakan untuk memindah cursor sesudahnya dan tombol Keluar digunakan untuk keluar dari form jadwal.

13. Perancangan Form Ubah Password

Lembar Data Ubah Pasword	
Password lama	<input type="text"/>
Password baru	<input type="text"/>
Ketik ulang Password	<input type="text"/>
<input type="button" value="OK"/>	<input type="button" value="Cencel"/>

Gambar 4.20 Desain Tampilan ubah password

Perancangan form password digunakan untuk menampilkan ubah password lama menjadi password baru, pada form ubah password terdapat 2 tombol yaitu tombol OK yang digunakan untuk mengganti password dan tombol Cencel digunakan untuk membatalkan penggantian password sekaligus kembali ke menu utama.

14. Perancangan Form Edit SetCom

SETUP	
Port	<input type="text"/>
Baud rate	<input type="text"/>
Data bits	<input type="text"/>
Stop bist	<input type="text"/>
Parity	None
Flow control	None
<input type="button" value="OK"/>	<input type="button" value="Cencel"/>

Gambar 4.21 Desain Tampilan SetCom

Perancangan form setup digunakan untuk tampilan komunikasi dengan simulasi , pada form ubah password terdapat 2 tombol yaitu tombol OK yang digunakan untuk mengkomunikasikan alat dan tombol Cancel digunakan untuk membatalkan komunikasi sekaligus kembali ke menu utama.

BAB V

IMPLEMENTASI SISTEM DAN PENGUJIAN ALAT

5.1. Implementasi

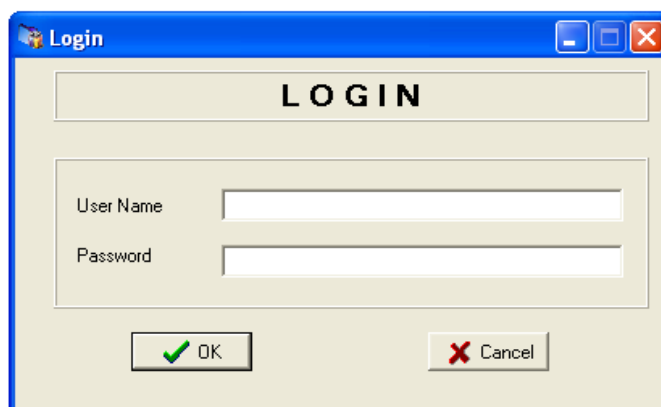
Berdasarkan hasil perancangan yang telah dilakukan pada bab IV, maka dalam implementasi ini akan menampilkan *form-form* yang terdapat pada program simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari.

5.1.1. Eksekusi Simulasi Sistem

Merupakan pembahasan tentang aplikasi program komponen yang dibutuhkan program, proses pada masing – masing program, cara kerja program dan penulisan *source code* program.

5.1.1.1. Tampilan Login

Merupakan tampilan awal program sebelum masuk ke menu utama, dimana kita diminta memasukan username dan password untuk proses login. Adapun Tampilan Login dapat dilihat pada gambar 5.1.



Gambar 5.1. *Tampilan Login*

1. Tombol Ok

```
procedure TFLOGIN.BitBtn2Click(Sender: TObject);  
begin  
if EUser.Text='ADMIN' then  
begin if EPassw.Text=dtpass then  
begin  
    FLOGIN.Hide;  
    MForm.Show;  
end  
else  
begin  
    ShowMessage('Data Password Salah');  
end; end  
else  
begin  
    ShowMessage('Bukan ADMIN');  
end; end;
```

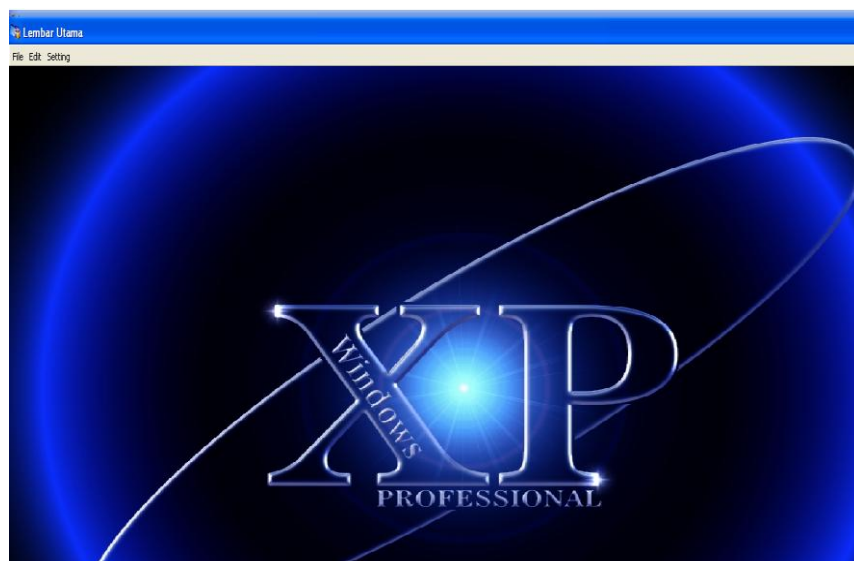
2. Tombol Cencel

```
procedure TFLOGIN.BitBtn1Click(Sender: TObject);  
begin  
    Application.Terminate;  
end; end.
```

5.1.1.2. Tampilan Menu Utama

Merupakan Tampilan setelah user berhasil melakukan proses login. Terdapat menu yang terdiri dari file, edit, setting. Pada menu File terdapat submenu yaitu : submenu data dosen yaitu untuk melihat data dosen yang sudah terdaftar, data mahasiswa yaitu untuk melihat data mahasiswa yang sudah terdaftar, data matakuliah yaitu untuk melihat data matakuliah yang sudah terdaftar, data jadwal yaitu untuk melihat data jadwal yang sudah dibuat, data masuk ruangan yaitu untuk melihat dokumentasi orang yang masuk ruangan, keluar yaitu untuk keluar program. Pada menu edit terdapat submenu edit dosen yaitu untuk melakukan proses input data dosen, edit mahasiswa yaitu untuk melakukan proses input data mahasiswa, edit

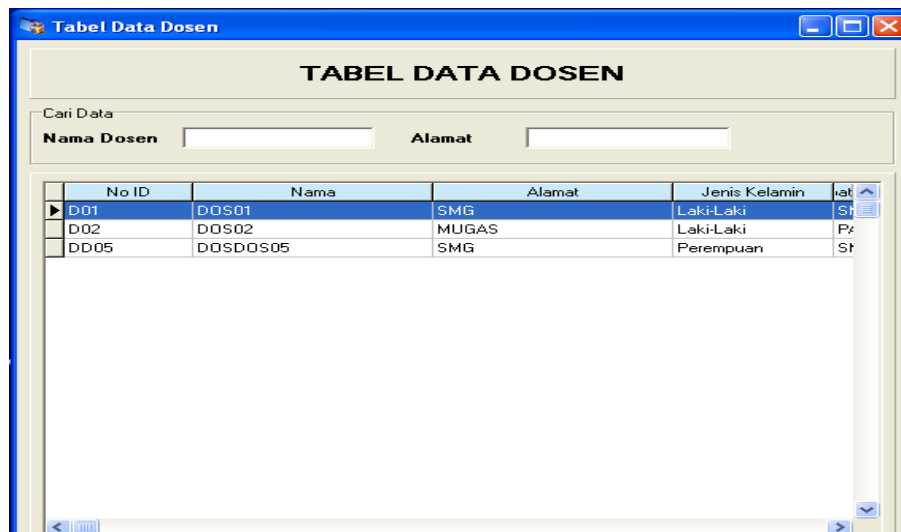
matakuliah yaitu untuk melakukan proses input data matakuliah, edit jadwal yaitu untuk melakukan proses input data jadwal. Pada menu setting terdapat submenu ganti password untuk mengganti password admin, sedangkan submenu setcom untuk menyambungkan komunikasi antara sistem penggunaan ruangan dengan simulasi penggunaan ruangan. Adapun tampilan Tampilan Menu Utama dapat dilihat pada gambar 5.2.



Gambar 5.2. *Tampilan Menu Utama*

5.1.1.3. Tampilan data dosen

Merupakan tampilan untuk melihat data dosen pengajar yang sudah terdata. Pada Tampilan ini terdapat 2 form pencarian yaitu form pencarian berdasarkan nama dosen dan form pencarian dosen berdasarkan alamat. Berikut tampilan Tampilan Ganti Password terdapat pada gambar 5.3.



Gambar 5.3. *Tampilan data dosen*

1. Tombol keluar

```
procedure TFtdosen.BitBtn1Click(Sender: TObject);
begin
    close;
end;
```

2. Pencarian Data Dosen

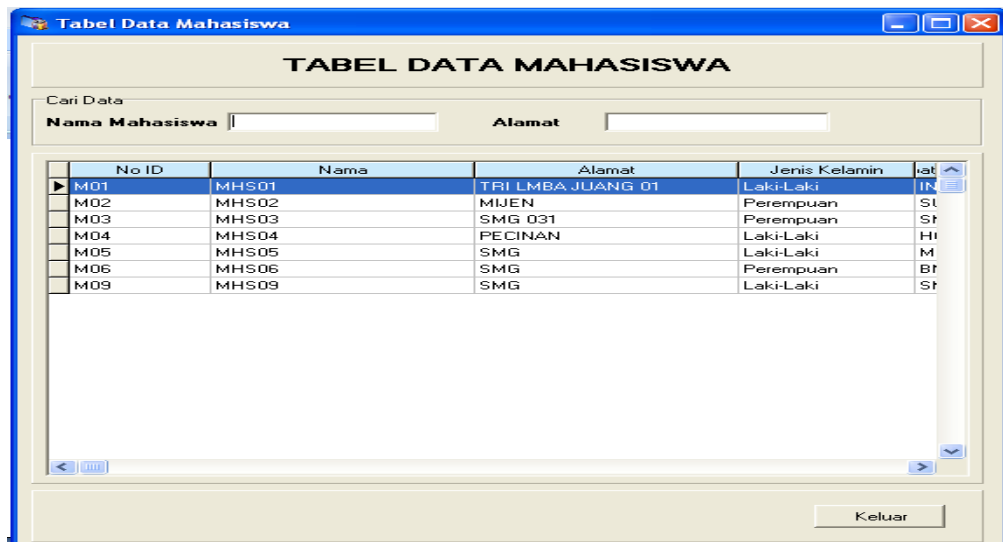
```
// Berdasarkan Nama Jadwal
procedure TFtdosen.ENmDosenKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    TDosen.Locate('nama',ENmDosen.Text,[lopartialkey]);
end;
```

```
// Berdasarkan Alamat
procedure TFtdosen.EAlmtKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    TDosen.Locate('alamat',EAlmt.Text,[lopartialkey]);
```

```
end;
```

5.1.1.4. Tampilan data mahasiswa

melihat data mahasiswa yang sudah terdata. Pada Tampilan ini terdapat 2 form pencarian yaitu form pencarian berdasarkan nama mahasiswa dan form pencarian mahasiswa berdasarkan alamat. Tampilan data mahasiswa terdapat pada gambar 5.4.



The screenshot shows a software window titled "Tabel Data Mahasiswa" with a search interface and a data table. The search interface includes two input fields: "Nama Mahasiswa" and "Alamat". The data table contains the following information:

No ID	Nama	Alamat	Jenis Kelamin	st
M01	MHS01	TRI LMBA JUANG 01	Laki-Laki	IK
M02	MHS02	MJEN	Perempuan	St
M03	MHS03	SMG 031	Perempuan	St
M04	MHS04	PECINAN	Laki-Laki	Hi
M05	MHS05	SMG	Laki-Laki	M
M06	MHS06	SMG	Perempuan	Bf
M09	MHS09	SMG	Laki-Laki	St

Gambar 5.4. Tampilan data mahasiswa

```

1. Tombol KELUAR
  procedure TFMhsiswa.BitBtn1Click(Sender: TObject);
  begin
  CLOSE;
  end;

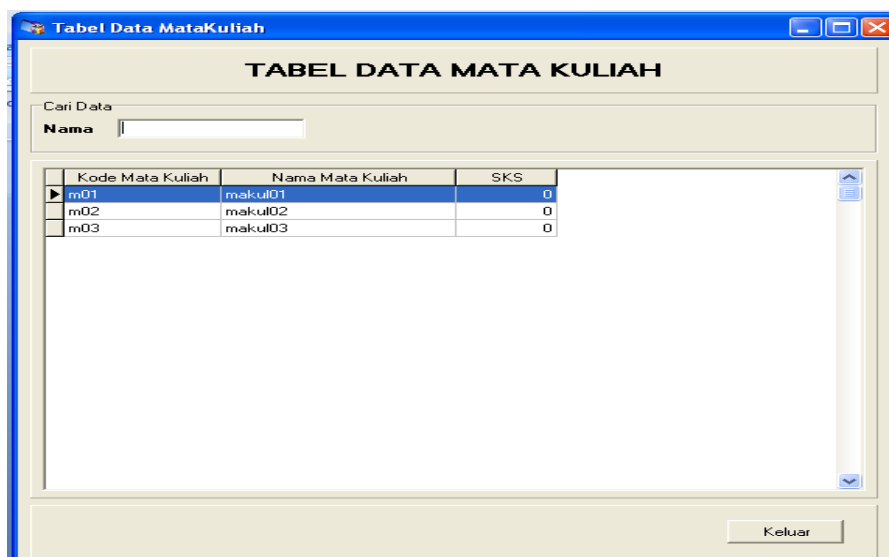
2. Pencarian Data Mahasiswa
  procedure TFMhsiswa.EnmKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
  begin
  Tmhsiswa.Locate('nama',Enm.Text,[lopartialkey]); // Berdasarkan Nama
  end;

  procedure TFMhsiswa.EalmtKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
  begin
  Tmhsiswa.Locate('alamat',Ealmt.Text,[lopartialkey]); // Berdasarkan
  Alamat
  end;

```

5.1.1.5. Tampilan Data matakuliah

Tampilan ini melihat data matakuliah. Pada Tampilan ini terdapat form pencarian yaitu form pencarian matakuliah berdasarkan nama matakuliah. Tampilan Data matakuliah terdapat pada gambar 5.5.



Gambar 5.5. Tampilan Data matakuliah

```

1. Tombol KELUAR
  procedure TFMtKuliah.BitBtn1Click(Sender: TObject);
  begin
    close;
  end;
2. Pencarian Matakuliah
  // Pencarian Berdasarkan Nama Matakuliah
  procedure TFMtKuliah.EnmKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  begin
    Tmtkul.Locate('nama',Enm.Text,[lopartialkey]);
  end;

```

5.1.1.6. Tampilan Data jadwal

Tampilan ini untuk melihat data jadwal penggunaan ruangan. Pada Tampilan ini terdapat 2 form pencarian yaitu form pencarian jadwal berdasarkan nama dosen dan form pencarian jadwal berdasarkan nama mahasiswa. Pada Tampilan ini terdapat tombol Keluar yaitu untuk kembali ke Menu Utama. Tampilan Data jadwal terdapat pada Gambar 5.6.

TABEL DATA JADWAL

Cari Data

Nama Dosen Nama Mahasiswa

Kode MAKUL	Nama MAKUL	Tanggal	Jam Masuk	Jam Selesai	NIK	Nama Dosen	Ruang	NIM
mtk01	matkul01	18/02/2013	15:40:11	16:40:11	D01	DOS01	1.Ruang A	M08
mtk01	matkul01	18/02/2013	15:40:11	16:40:11	D01	DOS01	1.Ruang A	M04
mtk01	matkul01	18/02/2013	15:40:11	16:40:11	D01	DOS01	1.Ruang A	M02
mtk01	matkul01	16/02/2013	9:42:59	11:00:59	D03	DOS03	1.Ruang A	M0E
mtk01	matkul01	16/02/2013	9:42:59	11:00:59	D03	DOS03	1.Ruang A	M03
mtk01	matkul01	16/02/2013	9:42:59	11:00:59	D03	DOS03	1.Ruang A	M01

Nama Dosen Nama Mahasiswa

noid	nama	kdmakul	nmmakul	alamat

noid	nama	kdmakul	nmmakul

Cetak Cetak Keluar

Gambar 5.6 *Tampilan Data jadwal*

1. Tombol KELUAR

```
procedure TFFTJadwal.BitBtn1Click(Sender: TObject);  
begin  
    close;  
end;
```

2. Tombol Cetak

```
procedure TFFTJadwal.BitBtn2Click(Sender: TObject);  
begin  
    FRepDosen.rep Dosen.Preview;  
end;  
procedure TFFTJadwal.cbIdMhsChange(Sender: TObject);  
begin  
    QMHSiswaMaKul.Close;  
    QMHSiswaMaKul.ParamByName('id').Value:=cbIdMhs.Text;  
    QMHSiswaMaKul.ExecSQL;  
    QMHSiswaMaKul.Open;  
end;  
procedure TFFTJadwal.BitBtn3Click(Sender: TObject);  
begin  
    FRepMhs.repMhs.Preview;  
end;  
end;
```

5.1.1.7. Tampilan Data masuk ruangan

Merupakan Tampilan untuk melihat dokumentasi masuk ruangan. Pada Tampilan ini terdapat 2 form pencarian yaitu form pencarian masuk ruangan berdasarkan nama dan form pencarian masuk ruangan berdasarkan status. Pada Tampilan ini terdapat tombol Keluar untuk kembali ke Menu Utama. Tampilan Data masuk ruangan seperti pada gambar 5.7.

Tabel Data Absen

TABEL DATA PENGGUNAAN RUANGAN

Cari Data

Nama **Status**

No ID	Nama	Ruang	Status
D01	DOS01	2.Ruang B	Pengajar
M02	MHS02	2.Ruang B	Mahasiswa
M06	MHS06	2.Ruang B	Mahasiswa
D03	DOS03	1.Ruang A	Pengajar
M01	MHS01	1.Ruang A	Mahasiswa
M03	MHS03	1.Ruang A	Mahasiswa
M04	MHS04	2.Ruang B	Mahasiswa

Keluar

Gambar 5.7. *Tampilan Data Masuk Ruangan*

```

1. Tombol KELUAR
  procedure TFTAbsen.BitBtn1Click(Sender: TObject);
  begin
    CLOSE;
  end;
2. Pencarian data masuk ruang
  procedure TFTAbsen.EnmKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  begin
    TAbsen.Locate('nama',ENm.Text,[lopartialkey]); // Berdasarkan Nama
  end;

  procedure TFTAbsen.EstaKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
  begin
    TAbsen.Locate('status',ESta.Text,[lopartialkey]); // Berdasarkan Status
  end;

```

5.1.1.8. Tampilan Edit Dosen

Merupakan Tampilan untuk menginput Data dosen. Pada Tampilan ini terdapat delapan tombol yaitu : tombol Lalu untuk menggeser cursor sebelumnya, tombol Berikut untuk menggeser cursor sesudahnya, tombol Ubah untuk mengubah data dosen yang sudah tersimpan , tombol Tambah untuk menambah data dosen, tombol Hapus untuk menghapus inputan data dosen yang salah, tombol Batal untuk membatalkan penyimpanan karena data dosen yang inputkan salah, tombol Simpan untuk menyimpan data dosen, dan tombol Keluar untuk kembali ke Menu Utama. Tampilan Data dosen seperti pada gambar 5.8.

Entry Data Dosen

No Induk Karyawan:

Nama Lengkap:

Foto diri:

Messages : None

Tempat dan Tanggal Lahir:

Agama:

Jenis Kelamin:

Alamat:

Status:

Mata Kuliah:

NIP	Nama Pengajar	Jenis Kelamin	Tempat Lahir	nggal L
D01	DOS01	Perempuan	S	31/08/20
D02	DOS02	Laki-Laki	D	31/08/20
D03	DOS03	Perempuan	S	31/08/20
D04	DOS04			31/08/20

ID Dosen	Nama Dosen	Kode Makul	Nama Makul
D1	DOSEN1	MK02	MAKJL02
D1	DOSEN1	MK04	MAKJL04
D1	DOSEN1	MK01	MAKJL011

Gambar 5.8. Tampilan Edit Dosen

```

1. Tombol Register Sidik Jari
procedure TFDosen.BRegFPClick(Sender: TObject);
var
    pErr: AErrors;
begin
    sampleNumber := 0;
    LInfo.Visible:=true;
    txtEvtMessage.Caption := 'Registration Mode';
    mode := 1; //In registration mode
    //DMFP.FPGetTemplate1.Disconnect;
    FPRegisterTemplate1.Run(1,pErr);
end;

2. Tombol Browse
procedure TFDosen.BitBtn2Click(Sender: TObject);
begin
    DBImage1DbClick(nil);
end;

3. Tombol Tambahkan
procedure TFDosen.SpeedButton1Click(Sender: TObject);
begin
    if dm.TMtKuliah.Locate('nama',CBMakul.Text,[locaseinsensitive])
    then
        begin
            dm.TMakuldsn.Open;
            if
                not(dm.TMakuldsn.Locate('iddosen;nmmakul',vararrayof([Dm.TDosennoide.Value,CBMakul.Text]),[locaseinsensitive])) then
                    begin
                        dm.TMakuldsn.Insert;
                        Dm.TMakuldsniddosen.Value:=Dm.TDosennoide.Value;
                        Dm.TMakuldsnnmdosen.Value:=Dm.TDosennama.Value;
                        Dm.TMakuldsnkdmakul.Value:=Dm.TMtKuliahkodemtk.Value;
                        Dm.TMakuldsnnmmakul.Value:=Dm.TMtKuliahnama.Value;
                        dm.TMakuldsn.Post;
                        Dm.MakulDsn.Close;
                        Dm.MakulDsn.Open;
                    end;
                end;
            end;
        end;
    end;

```

5.1.1.9. Tampilan Edit Mahasiswa

Merupakan Tampilan untuk input Data mahasiswa, Pada Tampilan ini terdapat delapan tombol yaitu : tombol Lalu untuk menggeser cursor sebelumnya,

tombol Berikut untuk menggeser cursor sesudahnya, tombol Ubah untuk mengubah data mahasiswa yang sudah tersimpan, tombol Tambah untuk menambah data mahasiswa, tombol Hapus untuk menghapus inputan data mahasiswa yang salah, tombol Batal untuk membatalkan penyimpanan karena data mahasiswa yang inputkan salah, tombol Simpan untuk menyimpan data mahasiswa, dan tombol Keluar untuk kembali ke Menu Utama. Tampilan edit mahasiswa dapat dilihat pada gambar 5.9.

The screenshot shows a web application window titled "Lembar Data Mahasiswa". The window is divided into two main sections: a form for editing student data on the left and a table of student records on the right.

Form Fields:

- No Induk Mahasiswa:
- Nama Lengkap:
- Register Sidik Jari:
- Foto diri: **Messages :** Data Sidik Jari Sudah Ada
- Jenis Kelamin:
- Tempat dan Tanggal Lahir:
- Agama:
- Alamat:
- Mata Kuliah:
- Dosen:

Table of Student Records:

NIM	Nama	Jenis Kelami	Tempat Lahir	Inggal La	Agama
M01	MHS01	Laki-Laki	H	31/08/200	ISLAM
M02	MHS02	Perempuan	G	31/08/200	KRISTE!
M03	MHS03	Perempuan	F	31/08/200	KATHOL
M04	MHS04	Perempuan	H	31/08/200	KRISTE!
M05	MHS05	Laki-Laki	J	31/08/200	KATHOL
M06	MHS06	Laki-Laki	K	31/08/200	KATHOL
M07	MHS07	Perempuan	H	31/08/200	KRISTE!

Buttons:

- Lalu, Tambah, Hapus, Simpan (top row)
- Berikut, Ubah, Batal (bottom row)
- Hapus, Keluar (bottom right)

Gambar 5.9. Tampilan edit mahasiswa

```

1. Tombol Register Sidik Jari
  procedure TFEDMhs.BRegFPClick(Sender: TObject);
  var
  pErr: AErrors;
  begin
    sampleNumber := 0;
    LInfo.Visible:=true;
    { fpImage1.Picture := nil;
    fpImage2.Picture := nil;
    fpImage3.Picture := nil;
    fpImage4.Picture := nil;}
    txtEvtMessage.Caption := 'Registration Mode';
    mode := 1; //In registration mode
    //btnVerify.Enabled := False;
    FPRegisterTemplate1.Run(1,pErr);
  end;
2. Tombol Browse
  procedure TFEDMhs.BitBtn2Click(Sender: TObject);
  begin
    DBImage1DblClick(nil);
  end;

```

5.1.1.10. Tampilan Edit Matakuliah

Merupakan Tampilan untuk menginput Data matakuliah, Pada Tampilan ini terdapat delapan tombol yaitu : tombol Lalu untuk menggeser cursor sebelumnya, tombol Berikut untuk menggeser cursor sesudahnya, tombol Ubah untuk mengubah data matakuliah yang sudah tersimpan, tombol Tambah untuk menambah data matakuliah, tombol Hapus untuk menghapus inputan data matakuliah yang salah, tombol Batal untuk membatalkan penyimpanan karena data matakuliah yang inputkan salah, tombol Simpan untuk menyimpan data matakuliah, dan tombol Keluar untuk kembali ke Menu Utama. Tampilan edit matakuliah terdapat pada gambar 5.10.

Kode Mata Kuliah: M04
Mata Kuliah: makul04
SKS: 2

Kode MTK	Nama Mata Kuliah	SKS
m01	makul01	0
m02	makul02	0
m03	makul03	0
M04	makul04	2

Lalu Ubah Hapus Simpan
Berikut Tambah Batal Keluar

Gambar 5.10. *Tampilan edit matakuliah*

1. Tombol Simpan

```

procedure TFDMTK.BSaveClick(Sender: TObject);
begin
if dm.TMtKuliah.State in [dsinsert,dsedit] then
begin
dm.TMtKuliah.Post;
end; end;

```

2. Tombol Batal

```

procedure TFDMTK.SpeedButton6Click(Sender: TObject);
begin
dm.TMtKuliah.Cancel;
end;

```

3. Tombol Tambah

```

procedure TFDMTK.BAddClick(Sender: TObject);
begin
dbekdmakul.SetFocus;
dm.TMtKuliah.Insert;
end;

```

5.1.1.11. Tampilan Edit Jadwal

Merupakan Tampilan untuk menginput Data jadwal, Pada Tampilan ini terdapat delapan tombol yaitu : tombol LALU untuk menggeser cursor sebelumnya, tombol BERIKUT untuk menggeser cursor sesudahnya, tombol UBAH untuk mengubah data jadwal yang sudah tersimpan, tombol HAPUS untuk menghapus inputan data jadwal yang salah, tombol BATAL untuk membatalkan penyimpanan karena data jadwal yang inputkan salah, tombol SIMPAN untuk menyimpan data jadwal, dan tombol KELUAR untuk kembali ke Menu Utama. berikut adalah tampilan edit jadwal terdapat pada gambar 5.11. :

The screenshot shows a software window titled "Lembar Data Jadwal". It features a form on the left for data entry, a central column of buttons, and two tables on the right.

Form Fields:

- Tanggal Kuliah: 12/02/2013
- Jam Masuk Kuliah: 10:52:36
- Jam Selesai Kuliah: 11:52:36
- Kode Mata Kuliah: [Dropdown]
- Mata Kuliah: [Text]
- ID Dosen: [Dropdown]
- Nama Dosen: [Text]
- Ruang: [Dropdown]
- NIM: 1406
- Nama Mahasiswa: 141506

Buttons: Lulu, Berikut, Ubah, Hapus, Batal, Simpan, Keluar.

Table 1 (Top Right):

NIP	Nama Dosen	Kd Makul	Nama Makul

Table 2 (Bottom Left):

Tanggal	Jam Masuk	Jam Selesai	Kode Mata Kuliah	Nama Mata Kuliah	NIP
12/02/2013	0:00:01	1:00:00	mik01	matkul01	D01
12/02/2013	0:00:01	1:00:00	mik01	matkul01	D01
12/02/2013	0:00:01	1:00:00	mik01	matkul01	D01
11/02/2013	23:30:01	23:59:00	mik01	matkul01	D01
11/02/2013	23:30:01	23:59:00	mik01	matkul01	D01
11/02/2013	23:30:01	23:59:00	mik01	matkul01	D01
11/02/2013	23:30:01	23:59:00	mik03	matkul03	D03
11/02/2013	23:30:01	23:59:00	mik03	matkul03	D03
11/02/2013	23:30:01	23:59:00	mik03	matkul03	D03
12/02/2013	0:00:01	1:00:00	mik03	matkul03	D03
12/02/2013	0:00:01	1:00:00	mik03	matkul03	D03

Table 3 (Bottom Right):

No Mahasi	Nama Mahasiswa	Kode makul	Nama Makul	ID Dosen	Nama Dosen

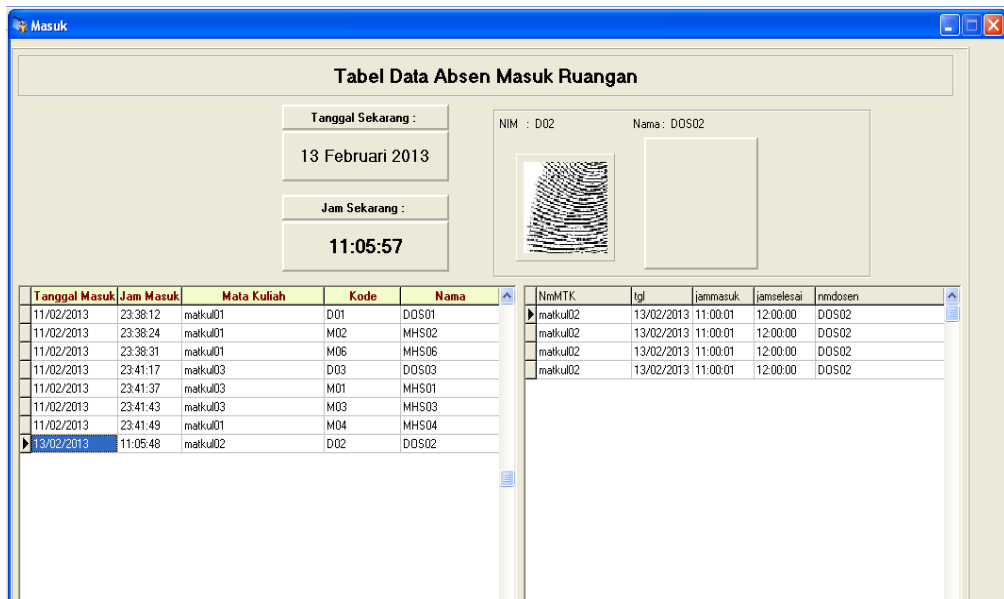
Gambar 5.11. Tampilan edit jadwal

1. Tombol Simpan

```
procedure TFDJadwal.BSaveClick(Sender: TObject);
begin
    TJadwalKul.Insert;
    TJadwalKultgl.Value:=ETgl.Date;
    TJadwalKuljammasuk.Value:=EJam.Time;
    TJadwalKulnim.Value:=tmhsmkdsnidmhs.Value;
    TJadwalKulnmmhsiswa.Value:=tmhsmkdsnmmhs.Value;
    TJadwalKuljamselesai.Value:=EJam2.Time;
    TJadwalKulkdmtk.Value:=EKdMaKul.Text;
    TJadwalKulNmMTK.Value:=EMaKul.Text;
    TJadwalKulnm Dosen.Value:=ENmDosen.Text;
    TJadwalKulnip.Value:=EKDDosen.Text;
    TJadwalKulruang.Value:=ERuang.Text;
    TJadwalKul.Post;
    tmhsmkdsn.Next;
end;
else
begin
    ShowMessage('Ruang '+ERuang.Text+' Sedang dipakai');
end;
```

5.1.1.12. Tampilan absen masuk ruang

Merupakan tampilan mengetahui siapa saja yang terabsensi dan memasuki ruangan berdasarkan jadwal. Berikut adalah tampilan absen masuk ruang terdapat pada gambar 5.12.



Gambar 5.12. *Tampilan Absen Masuk Ruang*

5.1.1.13. Tampilan ubah Password Admin

Merupakan tampilan untuk mengganti password Admin. Pada Tampilan ini terdapat 2 tombol yaitu tombol OK untuk mengganti password yang lama dengan yang baru dan tombol CENCEL untuk membatalkan proses sekaligus kembali ke Menu Utama. Berikut adalah tampilan ubah password terdapat pada gambar 5.13.



Gambar 5.13. *Tampilan Ubah Password*

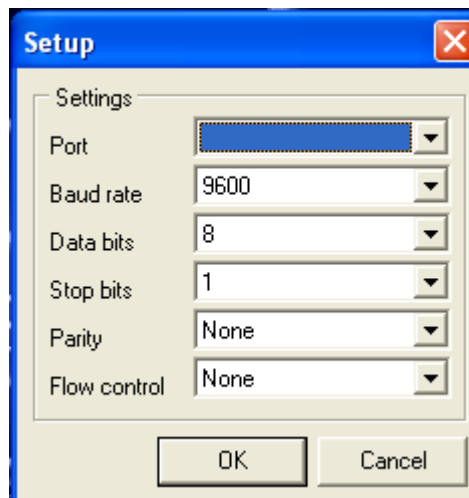
```

1. Tombol OK
procedure TFUbhPassw.BtnOKClick(Sender: TObject);
begin
  if (EPassBaru.Text<>"") or (EPassBaru2.Text<>"") then
  begin
    if EPassBaru.Text=EPassBaru2.Text then
    begin
      Memo1.Lines.Clear;
      Memo1.Lines.Add(EPassBaru.Text);
      Memo1.Lines.SaveToFile('.\dt.dat');
      ModalResult:=mrOk;
    end
    else
    begin
      ShowMessage('Data Password Salah !!!!');
      EPassBaru.SetFocus;
    end; end
  else
  begin
    ShowMessage('Data Password Tidak Boleh Kosong !!!!');
    EPassBaru.SetFocus;
  end; end;

```

5.1.1.14. Tampilan set com

Tampilan ini untuk menampilkan komunikasi data antara sistem dan simulasi penggunaan ruangan, pada tampilan set com ini terdapat dua tombol yaitu : tombol OK untuk melakukan komunikasi data, dan tombol CENCEL untuk membatalkan komunikasi data dan sekaligus kembali ke men utama. Tampilan Tindakan Penanganan terdapat pada gambar 5.13.



Gambar 5.14. *Tampilan set com*

5.2. Pengujian Alat (Hardwere)

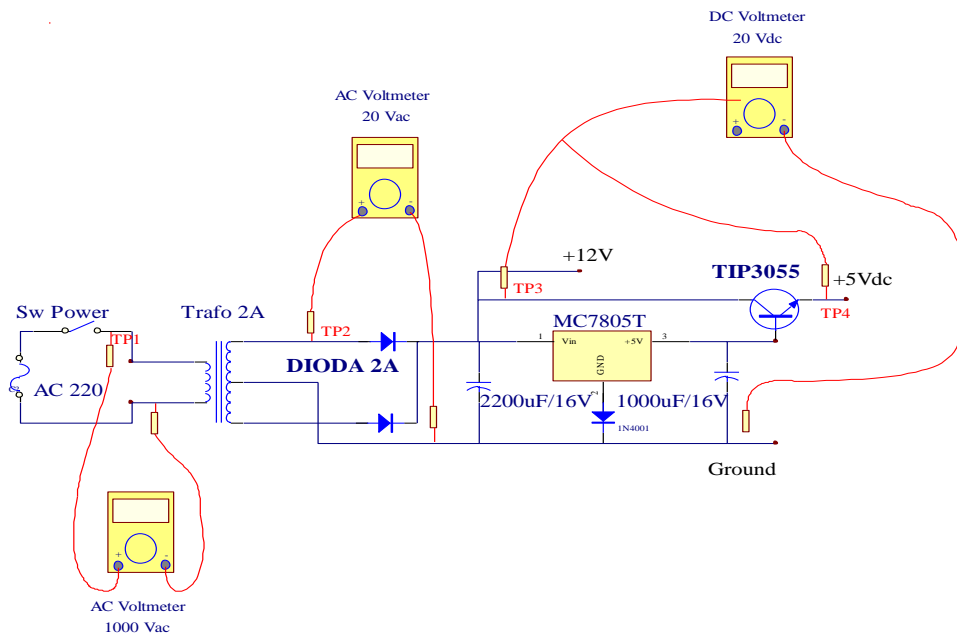
5.2.1. Power Supply

A. Langkah-langkah pengujian

Pengujian power supply dilakukan dengan langkah-langkah sebagai berikut:

1. Menyiapkan rangkaian catu daya yang akan diuji dan peralatan berupa multimeter.
2. Menghidupkan saklar catu daya (memberikan tegangan input 220 VAC pada input catu daya).
3. Melakukan pengukuran pada masing-masing keluaran catu daya.
4. Mencatat hasil pengukuran.

B. Hasil dan analisa pengujian



Gambar 5.15 Pengujian Power Supply

Dari pengujian dan pengukuran power supply diperoleh hasil:

Tabel 5.1 Hasil dan analisa

- a. Pengukuran tegangan input AC 220V (TP1)

Tegangan terukur	Tegangan semestinya	Error persen
210 Vac	220 Vac	4,5 %

eror : $((\text{tegangan terukur} - \text{tegangan semestinya}) / \text{Tegangan semestinya}) * 100\%$

: $((210 - 220) / 220) * 100\%$

: 4,5%

- b. Pengukuran tegangan sekunder trafo (TP2)

Tegangan terukur	Tegangan semestinya	Error persen
11,8 Vac	12 Vac	1,6 %

$$\begin{aligned} \text{eror} &: ((\text{tegangan terukur}-\text{tegangan semestinya})/\text{Tegangan semestinya}) * 100\% \\ &: ((11,8-12)/12) * 100\% \\ &: 1,6\% \end{aligned}$$

c. Pengukuran tegangan output penyearah (TP3)

Tegangan terukur	Tegangan semestinya	Error persen
11,3 Vdc	11,2 Vdc	0,89 %

$$\begin{aligned} \text{eror} &: ((\text{tegangan terukur}-\text{tegangan semestinya})/\text{Tegangan semestinya}) * 100\% \\ &: ((11,3-11,2)/11,2) * 100\% \\ &: 0,89\% \end{aligned}$$

d. Pengukuran tegangan output IC Regulator (TP4)

Tegangan terukur	Tegangan semestinya	Error persen
5,01 V	5,0 V	0,2 %

$$\begin{aligned} \text{eror} &: ((\text{tegangan terukur}-\text{tegangan semestinya})/\text{Tegangan semestinya}) * 100\% \\ &: ((5,01-5)/5) * 100\% \\ &: 0,2\% \end{aligned}$$

Dari hasil pengukuran tegangan keluaran catu daya 12 Volt mendekati sama dengan hasil pengukuran dengan menggunakan multimeter. Sedangkan pengukuran tegangan catu daya 5 Volt lebih 0,02 Volt, hal itu sudah berbeda dengan perencanaan sebesar 5 Volt, tetapi dengan tegangan 5,01 Volt sudah dapat digunakan untuk memberikan catu daya pada blok mikrokontroller.

5.2.2. PENGUJIAN MIKROKONTROLLER

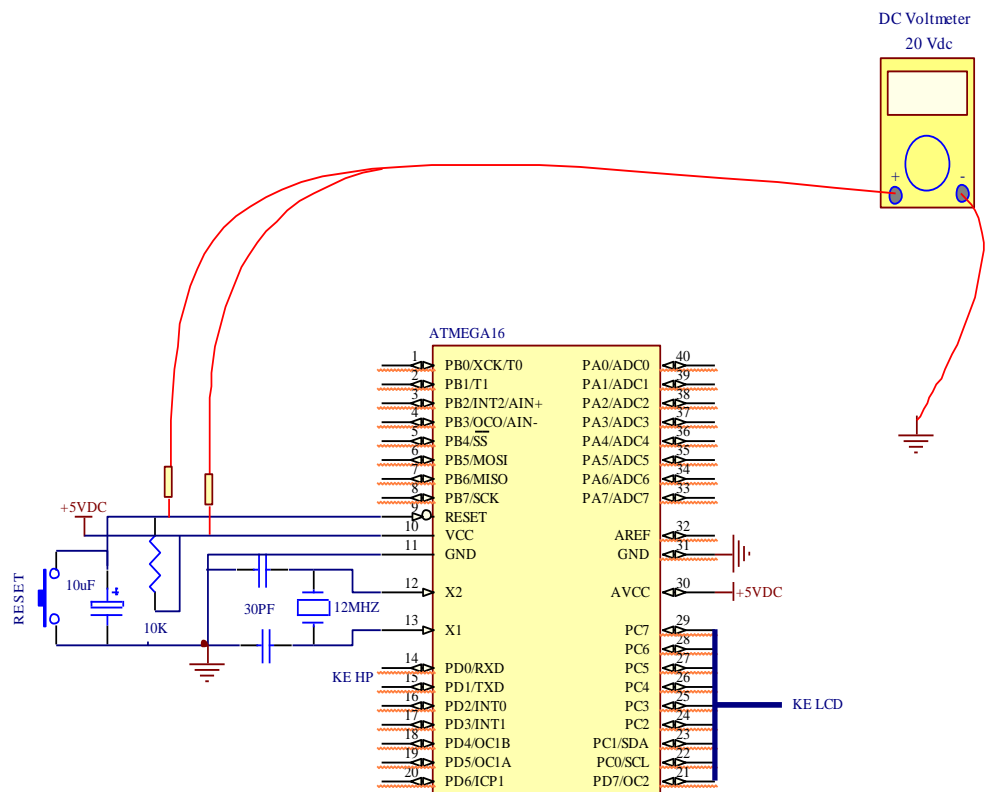
A. Langkah-Langkah Pengujian

Pengujian rangkaian mikrokontroller dilakukan dengan langkah-langkah sebagai berikut:

1. Menyiapkan rangkaian mikrokontroller yang akan diuji, rangkaian pendukung dan multimeter.
2. Menghubungkan rangkaian mikrokontroller dengan semua rangkaian sesuai gambar skema.
3. Menghidupkan saklar catu daya (memberikan tegangan input 220 VAC pada input catu daya).
4. Mencatat hasil pengujian dan menganalisa.

B. Hasil Dan Analisa Pengujian

Berdasarkan pada hasil pengujian dan pengukuran yang telah dilakukan, maka di dapat hasil sebagai berikut :



Gambar 5.16. Pengujian Mikrocontroller

- Pengukuran Pin VCC

Tabel 5.2 Hasil pengujian mikrokontroler

Tegangan terukur	Tegangan semestinya	Error persen
+4,85 V	5,0 V	3 %

eror : ((tegangan terukur-tegangan semestinya)/Tegangan semestinya)*100%

: ((4,85-5)/ 5)*100%

: 3%

- Pengukuran Pin Reset

Status sw	Tegangan pin reset	Keterangan
Ditekan	0,0 V	Mikro reset
Dilepas	4,82 V	Mikro jalan

- Pengukuran Port Mikro

Port	Logik High	Logik Low	Keterangan
A.0	4,80 V	0,0 V	Input
A.1	4,83 V	0,0 V	Input
A.2	4,83 V	0,0 V	Output
A.3	4,81 V	0,0 V	Output
A.4	4,85 V	0,0 V	Output
D.1	4,85 V	0,0 V	Output
D.2	4,85 V	0,0 V	Output
D.3	4,82 V	0,0 V	Output
D.4	4,87 V	0,0 V	Output
D.5	4,86 V	0,0 V	Output
D.6	4,83 V	0,0 V	Output
D.7	4,85 V	0,0 V	Output

4.2.3 DRIVER MOTOR PINTU

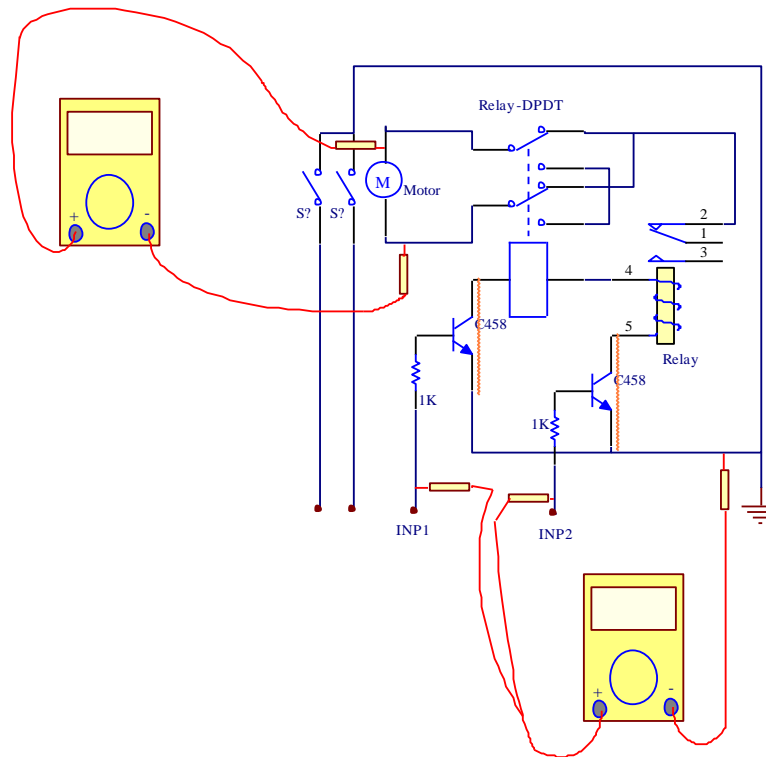
A. Langkah-langkah pengujian

Pengujian rangkaian drive motor pintu dilakukan dengan langkah-langkah sebagai berikut:

1. Menyiapkan rangkaian drive motor pintu yang akan diuji, rangkaian pendukung dan multimeter.
2. Menghubungkan rangkaian drive motor pintu dengan semua rangkaian sesuai gambar skema.
3. Menghidupkan saklar catu daya (memberikan tegangan input 220 VAC pada input catu daya).
4. Mencatat hasil pengujian dan menganalisa.

B. Hasil dan analisa pengujian

Berdasarkan pada hasil pengujian dan pengukuran yang telah dilakukan, maka di dapat hasil sebagai berikut :



Gambar 5.17. *Pengujian Drive Motor Pintu*

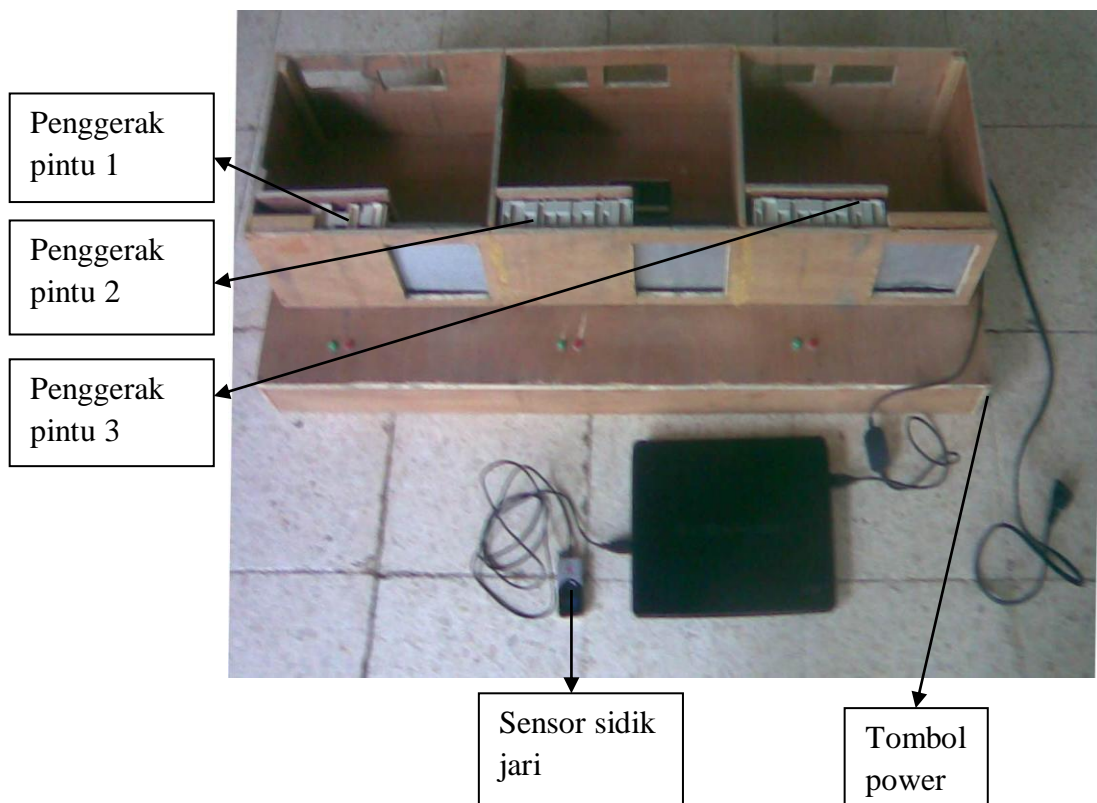
- Tabel pengukuran motor driver relay

Tabel 5.3 Hasil pengujian drive motor pintu

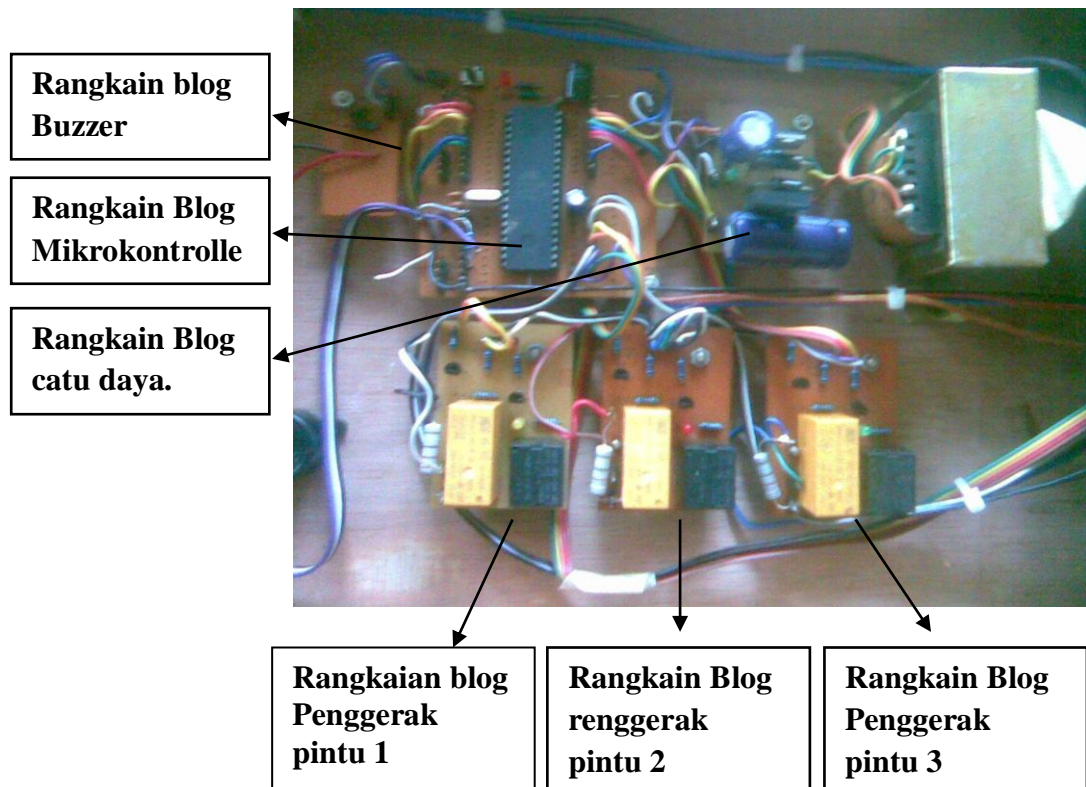
Inp1	Inp2	Tegangan motor	Keterangan
0,0 v	0,0 v	0 v	Motor mati
0,0 v	4,8 v	+10,8 v	Motor berputar ke kiri
4,8 v	4,8 v	-10,6 v	Motor berputar ke kanan

5.3. PENGUJIAN SISTEM SECARA KESELURUHAN

Setelah masing-masing rangkaian diuji dan sesuai dengan yang diharapkan, maka untuk langkah selanjutnya adalah menguji rangkaian keseluruhan dalam sebuah sistem yang terpadu.



Gambar 5.18. Bagian Luar Simulasi Sistem Penggunaan Ruang Berdasarkan Jadwal Dengan Menggunakan Sensor Sidik Jari.



Gambar 5.19. Bagian dalam Simulasi Sistem Penggunaan Ruang Berdasarkan Jadwal Dengan Menggunakan Sensor Sidik Jari

Adapun langkah-langkah yang harus dilakukan adalah sebagai berikut :

1. Menghubungkan rangkaian simulasi pada PC dan kabel power dengan sumber tegangan. Kemudian tombol power dinyalakan, sebagai tanda siap menerima input.
2. Dosen dan mahasiswa mengentri sidik jari sebagai identitas ID kedalam sistem penggunaan ruangan berdasarkan jadwal.
3. Mikro akan menerima inputan dari sistem setelah sensor sidik jari mendeteksi guratan sidik jari yang sudah dientri oleh mahasiswa maupun dosen. Kemudian mikro akan menggerakkan driver motor pintu dan menghidupkan buzzer.

BAB VI

PENUTUP

6.1 KESIMPULAN

Pada akhir bab ini penulis akan memberikan kesimpulan pada simulasi sistem yang telah selesai dibuat. Kesimpulan yang diambil adalah :

1. Dengan adanya simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari sangat membantu dalam penggunaan ruangan untuk kegiatan perkuliahan secara cepat dan tepat.
2. Dengan adanya simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari, sangat membantu dalam proses rekapitulasi siapa saja yang sudah masuk ruangan pada tiap periode dengan cepat dan tepat.
3. Didalam menggunakan simulasi sistem penggunaan ruangan berdasarkan jadwal, Tenaga yang digunakan untuk pengolahan data pengguna ruangan diminimalkan dan tidak memerlukan SDM terlalu banyak didalam pelaksanaannya, cukup dilakukan oleh satu orang.
4. Dengan adanya simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari dapat meminimalisasikan kesalahan perhitungan tercapai laporan, karena sistem komputer jauh lebih akurat dalam perhitungan di bandingkan tenaga manusia yang tergantung pada kondisi fisik dan mentalnya yang tidak selalu stabil.
5. Dari hasil pengujian yang dilakukan, diketahui bahwa komunikasi antara simulasi-simulasi sistem penggunaan ruangan dapat berjalan dengan baik.

6.2. SARAN

Dalam pengembangan sistem ini perlu adanya perbaikan dikemudian hari. Mungkin beberapa saran berikut dapat membantu dalam pengembangan sistem dikemudian hari :

1. simulasi sistem penggunaan ruangan ini masih sangat sederhana. Jika ingin lebih efektif dan efisien didalam penggunaannya mungkin lebih baik ditambahkan perangkat keras yang lebih maju didalamnya.
2. Program simulasi sistem penggunaan ruangan berdasarkan jadwal dengan menggunakan sensor sidik jari yang di buat sangatlah sederhana dan dapat dikembangkan lebih lanjut untuk menambah kemampuan didalam pemrosesan datanya.
3. Pengembangan simulasi sistem ini untuk kedepannya sangat diharapkan untuk membuat sistem database yang lebih kompleks agar dapat menyajikan informasi lebih banyak.

DAFTAR PUSTAKA

- Pranata, Anthony, 2002 *Pemrograman Borland Delphi 6 Edisi 4*, Andi, Yogyakarta.
- Husni, (2004), *Pemrograman DataBase Dengan Dhelpi*, Graha Ilmu, Yogyakarta
- Jogiyanto.HM, 2002, *Analisis dan Desain Sistem Informasi*, Andi Offset, Yogyakarta
- Kusnassriyanto dan Husni Iskandar Pohan, , 2004, *Pengantar perencanaan sistem*. Erlangga, Jakarta
- Kadir, Abdul, *Dasar Aplikasi Database MySQL – Delphi*, Penerbit Andi Yogyakarta, Yogyakarta, 2004
- Presman, Roger S, Ph. D., *Rekayasa Perangkat Lunak: Pendekatan Praktis (Buku Satu)*, Ed.:1, Penerbit Andi Yogyarta, Yogyakarta, 2002
- Teddy Marcus, Agus Prijono, Josep Widiadhi, (2004), *Delphi Developer Dan Sql Server 2000*, Informatika, Bandung
- Thomas j. Kakiay,2003, *Pengantar System Simulasi Edisi I*, Andi, Yogjakarta

<http://digilib.its.ac.id/public/ITS-Master-18059-2209206810-Paper.pdf>

diakses pada tanggal 19 November 2012.

Pembuatan Software Sistem Keamanan Pintu Dengan Barcode Reader Berbasis At89s51 menggunakan Visual Basic 6.0

<http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-ihsansania-15663> diakses pada tanggal 15 Desember 2012.

Pembuka pintu otomatis menggunakan avr atmega 8535 dan Sensor pir

<http://repository.gunadarma.ac.id/bitstream/123456789/1289/1/20107557.pdf>

diakses pada tanggal 04 January 2012.

Integrasi Pengendalian Pintu Masuk Ruang Dengan Sistem Informasi Data Dengan Bantuan Alat Pembaca Barcode

<http://123seminaronly.com/Seminar-Reports/2013-01/53642515-Bar-Code.pdf>

diakses pada tanggal 05 January 2012.

```

// LOGIN=====
unit ulogin;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, Mask, DBCtrls, db;
type
  TFLOGIN = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Mem1: TMemo;
    Label3: TLabel;
    EUser: TEdit;
    EPassw: TEdit;
    procedure FormShow(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  FLOGIN: TFLOGIN;
  dtnama, dtpass : string;
implementation
uses AbsU1, DMUnit, UPASSW, UFDTMhsw, UFDJdwAjar, UFDMTK,
UEntryDK, UMForm;
{$R *.dfm}
procedure TFLOGIN.FormShow(Sender: TObject);
begin
  Mem1.Clear;
  if FileExists('.\dt.dat') then
  begin
    Mem1.Lines.LoadFromFile('.\dt.dat');
    //dtnama:=Mem1.Lines[0];
    dtpass:=Mem1.Lines[0];
  end
  else
  begin
    //dtnama:='heri';
    dtpass:='789456';
    //Mem1.Lines.Append(dtnama);
  end;
end;

```



```

        Memol.Lines.Append(dtpass);
        Memol.Lines.SaveToFile('.\dt.dat');
    end; end;
procedure TFLOGIN.BitBtn2Click(Sender: TObject);
begin
    {FLOGIN.Hide;
MForm.Show;}
    if EUser.Text='ADMIN' then
    begin
        if EPassw.Text=dtpass then
        begin
            FLOGIN.Hide;
            MForm.Show;
        end
        else
        begin
            ShowMessage('Data Password Salah');
        end; end
    else
    begin
        ShowMessage('Bukan ADMIN');
    end; end;
procedure TFLOGIN.BitBtn1Click(Sender: TObject);
begin
    Application.Terminate;
end; end.

//EDIT MATAKULIAH=====

unit UFDMTK;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, Mask,
    DBCtrls, db;
type
    TFDMTK = class(TForm)
        Panell1: TPanel;
        Label1: TLabel;
        Label2: TLabel;
        dbekdmakul: TDBEdit;
        DBEdit4: TDBEdit;
        DBGrid1: TDBGrid;
        SpeedButton2: TSpeedButton;
        SpeedButton3: TSpeedButton;
        BAdd: TSpeedButton;
    end;

```

```

    BEdit: TSpeedButton;
    Bdel: TSpeedButton;
    SpeedButton6: TSpeedButton;
    BSave: TSpeedButton;
    Bevel1: TBevel;
    BitBtn1: TBitBtn;
    Label3: TLabel;
    DBEdit1: TDBEdit;
    procedure BSaveClick(Sender: TObject);
    procedure BAddClick(Sender: TObject);
    procedure BEditClick(Sender: TObject);
    procedure BdelClick(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    FDMTK: TFDMTK;
implementation
uses DMUnit;
{$R *.dfm}

procedure TFDMTK.BSaveClick(Sender: TObject);
begin
if dm.TMtKuliah.State in [dsinsert,dsedit] then
begin
    dm.TMtKuliah.Post;
end; end;
procedure TFDMTK.BAddClick(Sender: TObject);
begin
dbekdmakul.SetFocus;
dm.TMtKuliah.Insert;
end;
procedure TFDMTK.BEditClick(Sender: TObject);
begin
dbekdmakul.SetFocus;
dm.TMtKuliah.Edit;
end;
procedure TFDMTK.BdelClick(Sender: TObject);
begin
dm.TMtKuliah.Delete;
end;

```

```

procedure TFDMTK.SpeedButton6Click(Sender: TObject);
begin
dm.TMtKuliah.Cancel;
end;
procedure TFDMTK.BitBtn1Click(Sender: TObject);
begin
close;
end;
procedure TFDMTK.SpeedButton2Click(Sender: TObject);
begin
DM.TMtKuliah.Prior;
end;
procedure TFDMTK.SpeedButton3Click(Sender: TObject);
begin
DM.TMtKuliah.Next;
end; end.

//EDIT DOSEN=====

unit UEntryDK;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, DBCtrls,
ComCtrls,
Mask, ExtDlgs ,db, mySQLDbTables, OleServer,
DPSDKOPSLib_TLB,ActiveX,AXCtrls,DpSdkEngLib_TLB,
OleCtrls, comobj, IdGlobal, StrUtils;
type
TFDosen = class(TForm)
Panell: TPanel; Label1: TLabel; Label2: TLabel; Label3:
TLabel; Label5: TLabel; Label6: TLabel; Label8: TLabel; Label14:
TLabel; dbenik: TDBEdit; DBEdit4: TDBEdit; DBedit7: TDBEdit;
DBEdit10: TDBEdit;
DBMemo1: TDBMemo;
DBComboBox1: TDBComboBox;
ETglLahir: TDateTimePicker;
DBComboBox3: TDBComboBox;
Label4: TLabel;
DBImage1: TDBImage;
PicDialog: TOpenPictureDialog;
DBGrid1: TDBGrid;
BRegFP: TSpeedButton;
GroupBox1: TGroupBox;
fpImage1: TImage; fpImage2: TImage; fpImage3: TImage;
fpImage4: TImage;

```

```

txtMessage: TLabel; txtEvtMessage: TLabel;
FPRegisterTemplatel: TFPRegisterTemplate;
Label7: TLabel;
SpeedButton1: TSpeedButton;
CBMakul: TComboBox;
DBGrid2: TDBGrid;
TDosen: TmySQLTable;
TDosennoId: TStringField;
TDosennama: TStringField;
TDosenfoto: TBlobField;
TDosensidikjari: TBlobField;
TDosenjnskelamin: TStringField;
TDosentempatlahir: TStringField;
TDosentgllahir: TDateField;
TDosenagama: TStringField;
TDosenalamat: TStringField;
TDosenket: TStringField;
DSDosen: TDataSource;
LInfo: TLabel;
BitBtn2: TBitBtn;
Panel2: TPanel;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
Badd: TSpeedButton;
Bedit: TSpeedButton;
SpeedButton6: TSpeedButton;
Bdel: TSpeedButton;
Bsave: TSpeedButton;
Panel3: TPanel;
BitBtn1: TBitBtn;
SpeedButton4: TSpeedButton;
procedure FormClose(Sender: TObject; var Action:
TCloseAction);
procedure BitBtn1Click(Sender: TObject);
procedure BsaveClick(Sender: TObject);
procedure DBImage1DbClick(Sender: TObject);
procedure BaddClick(Sender: TObject);
procedure BeditClick(Sender: TObject);
procedure BdelClick(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure BRegFPClick(Sender: TObject);
procedure FPRegisterTemplatelDevConnected(Sender: TObject);
procedure FPRegisterTemplatelDevDisconnected(Sender:
TObject);
procedure FPRegisterTemplatelDone(Sender: TObject;

```

```

        var pTemplate: OleVariant);
    procedure FPRegisterTemplatelError(Sender: TObject; errcode:
TOleEnum);
    procedure FPRegisterTemplatelSampleQuality(Sender: TObject;
    Quality: TOleEnum);
    procedure FPRegisterTemplatelSampleReady(Sender: TObject;
    var pSample: OleVariant);
    procedure FormShow(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    FDosen: TFDosen;
    sampleNumber: Integer;
    fpData : array[0..2047] of Byte;
    blobSize: Integer;
    mode : Integer;
    regtemplate : fptemplate;
    v : variant;
    blob : olevariant;
    bufvarray:Pbytearray;
    perr : Aierrors;
implementation
uses DMUnit, UExtFP;
{$R *.dfm}
procedure TFDosen.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    FPRegisterTemplatel.Disconnect;
    DMFP.FPGetTemplatel.Run(1,perr);
    ModalResult:=mrok;
end;
procedure TFDosen.BitBtn1Click(Sender: TObject);
begin
    close;
end;
procedure TFDosen.BsaveClick(Sender: TObject);
begin
    LInfo.Visible:=false;
    if dm.TDosen.State in [dsedit,dsinsert] then
    begin
        Dm.TDosentgllahir.Value:=ETgllahir.Date;
    end;
end;

```

```

    Dm.TDosen.Post;
    BRegFP.Enabled:=false;
        fpImage1.Picture:=nil;
        fpImage2.Picture:=nil;
        fpImage3.Picture:=nil;
        fpImage4.Picture:=nil;
    end;
    {if BSimpan.Tag=0 then
    begin
    if MessageDlg('Tambah Data Lagi
?',mtConfirmation,[mbyes,mbcancel],0)=mryes then
        begin
            Dm.TDosen.Insert;
            DBEdit2.SetFocus;
            end
            else close;
        end
        else close;}
end;
procedure TFDosen.DBImage1DbClick(Sender: TObject);
begin
    if Dm.TDosen.State in [dsedit,dsinsert] then
        begin
            if PicDialog.Execute then
                begin
                    Dm.TDosenfoto.LoadFromFile(PicDialog.FileName);
                end; end; end;
end;
procedure TFDosen.BaddClick(Sender: TObject);
begin
    if Dm.TDosen.State=dsbrowse then
        begin
            dbenik.SetFocus;
            Dm.TDosen.Insert;
            BRegFP.Enabled:=true;
            end; end;
end;
procedure TFDosen.BeditClick(Sender: TObject);
begin
    dbenik.SetFocus;
    dm.TDosen.Edit;
end;
procedure TFDosen.BdelClick(Sender: TObject);
begin
    if not(Dm.TDosen.IsEmpty) then
        begin
            dm.TDosen.Delete;
            fpImage1:=nil;
            fpImage2:=nil ;
        end;
    end;
end;

```

```

    fpImage3:=nil;
    fpImage4:=nil;
end; end;
procedure TFDosen.SpeedButton6Click(Sender: TObject);
begin
    LInfo.Visible:=false;
    fpImage1:=nil;
    fpImage2:=nil;
    fpImage3:=nil;
    fpImage4:=nil;
    FPRegisterTemplatel.Disconnect;
    dm.TDosen.Cancel;
    BRegFP.Enabled:=false;
end;
procedure TFDosen.SpeedButton2Click(Sender: TObject);
begin
    dm.TDosen.Prior;
end;
procedure TFDosen.SpeedButton3Click(Sender: TObject);
begin
    dm.TDosen.Next;
end;
procedure TFDosen.BRegFPClick(Sender: TObject);
var
    pErr: AIErrors;
begin
    sampleNumber := 0;
    LInfo.Visible:=true;
    { if not(fpImage1.Picture = nil) then fpImage1.Picture := nil;
      if not(fpImage2.Picture = nil) then fpImage2.Picture := nil;
      if not(fpImage3.Picture = nil) then fpImage3.Picture := nil;
      if not(fpImage4.Picture = nil) then fpImage4.Picture := nil;}
    txtEvtMessage.Caption := 'Registration Mode';
    mode := 1; //In registration mode
    //DMFP.FPGetTemplatel.Disconnect;
    FPRegisterTemplatel.Run(1,pErr);
end;
procedure TFDosen.FPRegisterTemplatelDevConnected(Sender:
TObject);
begin
    txtEvtMessage.Caption := 'Sensor Connected !!';
    sleep(1000);
    if mode = 1 then
        txtEvtMessage.Caption := ' Continue Registration Mode !!';
end;
procedure TFDosen.FPRegisterTemplatelDevDisconnected(Sender:
TObject);

```

```

begin
    txtEvtMessage.Caption := 'Sensor Disconnected !!';
end;
procedure TFDosen.FPRegisterTemplate1Done(Sender: TObject;
    var pTemplate: OleVariant);
var
    l_template : IFPTemplate;
    l_dispatch1 : IDispatch;
    idx : word;
    blobstm : tstream;
begin
    l_dispatch1 := IDispatch(pTemplate);
    l_dispatch1.QueryInterface(IID_IFPTemplate, l_template);
    blob := VarArrayCreate([0, 2047], varByte);
    l_template.Export(blob);
    dm.TDosen.Edit;

blobstm:=dm.TDosen.CreateBlobStream(dm.TDosenSidikjari, bmReadWrite);
    blobstm.Size;
    bufvarray:=VarArrayLock(blob);
    for idx:=0 to 2047 do
    begin
        fpData[idx]:=bufvarray[idx];
    end;
    VarArrayUnlock(blob);
    blobstm.Seek(0, 0);
    blobstm.Write(fpdata, 2047);
    blobSize:=2047;
    blobstm.Free;
end;
procedure TFDosen.FPRegisterTemplate1Error(Sender: TObject;
    errcode: TOleEnum);
begin
    txtEvtMessage.Caption := 'Error !!';
end;
procedure TFDosen.FPRegisterTemplate1SampleQuality(Sender:
TObject;
    Quality: TOleEnum);
begin
    if Quality <> Sq_Good then
    begin
        txtEvtMessage.Caption := 'Image Quality Not good !!';
        sampleNumber := sampleNumber -1;
    end end;
end;
procedure TFDosen.FPRegisterTemplate1SampleReady(Sender:
TObject;

```



```

    var pSample: OleVariant);
var
    l_sample : IFPSample;
    l_dispatch1 : IDispatch;
    l_dispatch2 : IDispatch;
begin
    l_dispatch1 := IDispatch(pSample);
    l_dispatch1.QueryInterface(IID_IFPSample, l_sample);
    if sampleNumber = 0 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage1.Width);
        l_sample.Set_PictureHeight(fpImage1.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage1.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #1';
    end;
    if sampleNumber = 1 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage2.Width);
        l_sample.Set_PictureHeight(fpImage2.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage2.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #2';
    end ;
    if sampleNumber = 2 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage3.Width);
        l_sample.Set_PictureHeight(fpImage3.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage3.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #3';
    end;
    if sampleNumber = 3 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage4.Width);
        l_sample.Set_PictureHeight(fpImage4.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage4.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #4';
    end;
end;

```

```

    end;
    sampleNumber := sampleNumber + 1;
end;
procedure TFDosen.FormShow(Sender: TObject);
begin
    CBMakul.Items.Clear;
    Dm.TMtKuliah.Open;
    Dm.TMtKuliah.First;
    while not(Dm.TMtKuliah.Eof) do
    begin
        CBMakul.Items.Add(Dm.TMtKuliahnama.Value);
        Dm.TMtKuliah.Next;
    end; end;
procedure TFDosen.SpeedButton1Click(Sender: TObject);
begin
    if
dm.TMtKuliah.Locate('nama',CBMakul.Text,[locaseinsensitive])
then
    begin
        dm.TMakuldsn.Open;
        if
not(dm.TMakuldsn.Locate('iddosen;nmmakul',vararrayof([Dm.TDosenno
oid.Value,CBMakul.Text]),[locaseinsensitive])) then
        begin
            dm.TMakuldsn.Insert;
            Dm.TMakuldsnidosen.Value:=Dm.TDosenno
            Dm.TMakuldsnnmdosen.Value:=Dm.TDosennama.Value;
            Dm.TMakuldsnkdmakul.Value:=Dm.TMtKuliahkode
            Dm.TMakuldsnnmmakul.Value:=Dm.TMtKuliahnama.Value;
            dm.TMakuldsn.Post;
            Dm.MakulDsn.Close;
            Dm.MakulDsn.Open;
        end; end; end;
procedure TFDosen.BitBtn2Click(Sender: TObject);
begin
    DBImage1Db1Click(nil);
end;
procedure TFDosen.SpeedButton4Click(Sender: TObject);
begin
    if not(Dm.MakulDsn.IsEmpty) then
    begin
        dm.MakulDsn.Delete;
    end; end; end.

// EDIT MAHASISWA

unit UFDTMhsw;

```

```

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, DBCtrls,
  ComCtrls,
  Mask, ExtDlgs ,db, mySQLDbTables, OleServer,
  DPSDKOPSLib_TLB,ActiveX,AXCtrls,DpSdkEngLib_TLB,
  OleCtrls, comobj;
type
  TFEDMhs = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label8: TLabel;
    Label4: TLabel;
    dbenama: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit7: TDBEdit;
    DBMemo1: TDBMemo;
    DBComboBox1: TDBComboBox;
    ETglLahir: TDateTimePicker;
    DBComboBox3: TDBComboBox;
    DBImage1: TDBImage;
    DBGrid1: TDBGrid;
    PicDialog: TOpenPictureDialog;
    QMTKMHS: TmySQLQuery;
    DSQMTKMHS: TDataSource;
    TMhsw: TmySQLTable;
    DSMhsw: TDataSource;
    GroupBox1: TGroupBox;
    fpImage1: TImage;
    fpImage2: TImage;
    fpImage3: TImage;
    fpImage4: TImage;
    BRegFP: TSpeedButton;
    txtMessage: TLabel;
    txtEvtMessage: TLabel;
    FPRegisterTemplatel: TFPRegisterTemplate;
    TMhswnoid: TStringField;
    TMhswnama: TStringField;
    TMhswfoto: TBlobField;
    TMhswsidikjari: TBlobField;
    TMhswjnskelamin: TStringField;
  end;

```

```
TMhswtempatlahir: TStringField;
TMhswtgllahir: TDateField;
TMhswagama: TStringField;
TMhswalamat: TStringField;
TMhswket: TStringField;
Label7: TLabel;
SpeedButton1: TSpeedButton;
CBMakul: TComboBox;
Label9: TLabel;
CBIdDosen: TComboBox;
SpeedButton4: TSpeedButton;
getid: TMySQLQuery;
dsgetid: TDataSource;
DBGrid2: TDBGrid;
DBGrid3: TDBGrid;
MakulMhs: TMySQLQuery;
DSMakulMhs: TDataSource;
MakulMhsidmhs: TStringField;
MakulMhsnmmhs: TStringField;
MakulMhskdmakul: TStringField;
MakulMhsnmmakul: TStringField;
MakulMhsiddosen: TStringField;
MakulMhsnmdosen: TStringField;
BitBtn2: TBitBtn;
LInfo: TLabel;
Panel2: TPanel;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
BAdd: TSpeedButton;
Bedit: TSpeedButton;
Bdel: TSpeedButton;
SpeedButton6: TSpeedButton;
Bsave: TSpeedButton;
Panel3: TPanel;
BitBtn1: TBitBtn;
SpeedButton5: TSpeedButton;
procedure BeditClick(Sender: TObject);
procedure BAddClick(Sender: TObject);
procedure BdelClick(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure BsaveClick(Sender: TObject);
procedure DBImage1Db1Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure TMhswAfterScroll(DataSet: TDataSet);
procedure FPRegisterTemplate1DevConnected(Sender: TObject);
```

```

    procedure FPRegisterTemplatelDevDisconnected(Sender:
TObject);
    procedure FPRegisterTemplatelDone(Sender: TObject;
    var pTemplate: OleVariant);
    procedure FPRegisterTemplatelError(Sender: TObject; errcode:
TOleEnum);
    procedure FPRegisterTemplatelSampleQuality(Sender: TObject;
    Quality: TOleEnum);
    procedure FPRegisterTemplatelSampleReady(Sender: TObject;
    var pSample: OleVariant);
    procedure BRegFPClick(Sender: TObject);
    procedure FormHide(Sender: TObject);
    procedure TMhswBeforeDelete(DataSet: TDataSet);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure CBMakulChange(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    FEDMhs: TFEDMhs;
    sampleNumber: Integer;
    fpData : array[0..2047] of Byte;
    blobSize: Integer;
    mode : Integer;
    regtemplate : fptemplate;
    v : variant;
    blob : olevariant;
    bufvarray:Pbytearray;
    kdmhsact : string;
implementation
uses DMUnit, AbsU1, UExtFP;
{$R *.dfm}
procedure TFEDMhs.BeditClick(Sender: TObject);
begin
    dbenama.SetFocus;
    BRegFP.Enabled:=true;
    TMhsw.Edit;
end;

```

```

procedure TFEDMhs.BAddClick(Sender: TObject);
begin
dbenama.SetFocus;
BRegFP.Enabled:=true;
TMhsw.Insert;
end;
procedure TFEDMhs.BdelClick(Sender: TObject);
begin
  IF not(TMhsw.IsEmpty) then
  begin
    TMhsw.Delete;
  end; end;
procedure TFEDMhs.SpeedButton6Click(Sender: TObject);
begin
LInfo.Visible:=False;
BRegFP.Enabled:=false;
TMhsw.Cancel;
end;
procedure TFEDMhs.BsaveClick(Sender: TObject);
begin
  LInfo.Visible:=False;
  BRegFP.Enabled:=false;
  if TMhsw.State in [dsedit,dsinsert] then
  begin
    TMhswtglLahir.Value:=ETglLahir.Date;
    TMhsw.Post;
    BRegFP.Enabled:=false;
    fpImage1.Picture:=nil;
    fpImage2.Picture:=nil;
    fpImage3.Picture:=nil;
    fpImage4.Picture:=nil;
  end; end;
procedure TFEDMhs.DBImage1Db1Click(Sender: TObject);
begin
  if TMhsw.State in [dsedit,dsinsert] then
  begin
    if PicDialog.Execute then
    begin
      TMhswfoto.LoadFromFile(PicDialog.FileName);
    end; end;end;
procedure TFEDMhs.BitBtn1Click(Sender: TObject);
begin
  CLOSE;
end;
procedure TFEDMhs.FormShow(Sender: TObject);
begin
  DMFP.FPGetTemplatel1.Disconnect;

```

```

sleep(1000);
TMhsw.Open;
MakulMhs.Open;
CBMakul.Items.Clear;
Dm.TMtKuliah.Open;
Dm.TMtKuliah.First;
while not(Dm.TMtKuliah.Eof) do
begin
    CBMakul.Items.Add(Dm.TMtKuliahkodevtk.Value);
    Dm.TMtKuliah.Next;
end; end;
procedure TFEDMhs.SpeedButton7Click(Sender: TObject);
begin
{   Dm.TMTKMHS.Insert;
    dm.TMTKMHSkdmhs.Value:=TMhswnoid.Value;
    dm.TMTKMHSnamamhs.Value:=TMhswnama.Value;
    dm.TMTKMHSnmmakul.Value:=dm.TMtKuliahnama.Value;
    dm.TMTKMHS.Post;
    QMTKMHS.Close;
    QMTKMHS.SQL.Clear;
    QMTKMHS.SQL.Add('select * from makulmhs where(kdmhs=:kdmhs)');
    QMTKMHS.Prepare;
    QMTKMHS.ParamByName('kdmhs').Value:=TMhswnoid.Value;
    QMTKMHS.ExecSQL;
    QMTKMHS.Open;
}
end;
procedure TFEDMhs.TMhswAfterScroll(DataSet: TDataSet);
begin
    {QMTKMHS.Close;
    QMTKMHS.SQL.Clear;
    QMTKMHS.SQL.Add('select * from makulmhs where(kdmhs=:kdmhs)');
    QMTKMHS.Prepare;
    QMTKMHS.ParamByName('kdmhs').Value:=TMhswnoid.Value;
    QMTKMHS.ExecSQL;}
    if MakulMhs.Active then
    begin
        MakulMhs.Close;
        MakulMhs.SQL.Clear;
        MakulMhs.SQL.Add('SELECT * FROM TMakulMhs K where
(K.idmhs=:idmhs)');
        MakulMhs.ParamByName('idmhs').Value:=TMhswnoid.Value;
        MakulMhs.ExecSQL;
        MakulMhs.Open;
    end;
    if TMhsw.State in [dsbrowse] then

```

```

begin
    if TMhswsidikjari.BlobSize>0 then
        txtEvtMessage.Caption:='Data Sidik Jari Sudah Ada'
    else
        txtEvtMessage.Caption:='Data Sidik Jari Belum Ada';
    end; end;
procedure TFEDMhs.FPRegisterTemplatelDevConnected(Sender:
TObject);
begin
    txtEvtMessage.Caption := 'Sensor Connected !!!';
    sleep(1000);
    if mode = 1 then
        txtEvtMessage.Caption := ' Continue Registration Mode !!!';
end;
procedure TFEDMhs.FPRegisterTemplatelDevDisconnected(Sender:
TObject);
begin
    txtEvtMessage.Caption := 'Sensor Disconnected !!!';
end;
procedure TFEDMhs.FPRegisterTemplatelDone(Sender: TObject;
var pTemplate: OleVariant);
var
    l_template : IFPTemplate;
    l_dispatch1 : IDispatch;
    idx : word;
    blobwrt : array[0..2047] of byte;
    blobstm : tstream;
    filestm : tfilestream;
begin
    l_dispatch1 := IDispatch(pTemplate);
    l_dispatch1.QueryInterface(IID_IFPtemplate,l_template);
    blob := VarArrayCreate([0,2047],varByte);
    l_template.Export(blob);
    TMhsw.Edit;
    blobstm:=TMhsw.CreateBlobStream(TMhswsidikjari,bmReadWrite);
//    filestm:=TFileStream.Create('.\datafp.txt',fmCreate);
    blobstm.Size;
    bufvarray:=VarArrayLock(blob);
    for idx:=0 to 2047 do
        begin
            fpData[idx]:=bufvarray[idx];
        end;
    VarArrayUnlock(blob);
    blobstm.Seek(0,0);
    blobstm.Write(fpdata,2047);
//    filestm.Seek(0,0);
//    filestm.Write(fpdata,2047);

```



```

// TMhsw.Post;
blobSize:=2047;
blobstm.Free;
// filestm.Free;
//btnVerify.Enabled:=true;
end;
procedure TFEDMhs.FPRegisterTemplatelError(Sender: TObject;
errcode: TOleEnum);
begin
txtEvtMessage.Caption := 'Error !!';
end;
procedure TFEDMhs.FPRegisterTemplatelSampleQuality(Sender:
TObject;
Quality: TOleEnum);
begin
if Quality <> Sq_Good then
begin
txtEvtMessage.Caption := 'Image Quality Not good !!';
sampleNumber := sampleNumber -1;
end end;
procedure TFEDMhs.FPRegisterTemplatelSampleReady(Sender:
TObject;
var pSample: OleVariant);
var
l_sample : IFPSample;
l_dispatch1 : IDispatch;
l_dispatch2 : IDispatch;
begin
l_dispatch1 := IDispatch(pSample);
l_dispatch1.QueryInterface(IID_IFPSample,l_sample);
if sampleNumber = 0 then
begin
l_sample.Set_PictureOrientation(1);
l_sample.Set_PictureWidth (fpImage1.Width);
l_sample.Set_PictureHeight(fpImage1.Height);
l_dispatch2 := l_sample.Get_Picture;
SetOlePicture(fpImage1.Picture,
IPictureDisp(l_dispatch2));
txtEvtMessage.Caption := 'Image #1';
end;
if sampleNumber = 1 then
begin
l_sample.Set_PictureOrientation(1);
l_sample.Set_PictureWidth (fpImage2.Width);
l_sample.Set_PictureHeight(fpImage2.Height);
l_dispatch2 := l_sample.Get_Picture;

```

```

        SetOlePicture(fpImage2.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #2';
    end ;
    if sampleNumber = 2 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage3.Width);
        l_sample.Set_PictureHeight(fpImage3.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage3.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #3';
    end;
    if sampleNumber = 3 then
    begin
        l_sample.Set_PictureOrientation(1);
        l_sample.Set_PictureWidth (fpImage4.Width);
        l_sample.Set_PictureHeight(fpImage4.Height);
        l_dispatch2 := l_sample.Get_Picture;
        SetOlePicture(fpImage4.Picture,
IPictureDisp(l_dispatch2));
        txtEvtMessage.Caption := 'Image #4';
    end;
    sampleNumber := sampleNumber + 1;
end;
procedure TFEDMhs.BRegFPClick(Sender: TObject);
var
pErr: AErrors;
begin
    sampleNumber := 0;
    LInfo.Visible:=true;
    { fpImage1.Picture := nil;
    fpImage2.Picture := nil;
    fpImage3.Picture := nil;
    fpImage4.Picture := nil;}
    txtEvtMessage.Caption := 'Registration Mode';
    mode := 1; //In registration mode
    //btnVerify.Enabled := False;
    FPRegisterTemplatel.Run(1,pErr);
end;
procedure TFEDMhs.FormHide(Sender: TObject);
begin
    FPRegisterTemplatel.Disconnect;
end;
procedure TFEDMhs.TMhswBeforeDelete(DataSet: TDataSet);
begin

```

```

    kdmhsact:=TMhswnoid.Value;
    //QMTKMHS.Open;
end;
procedure TFEDMhs.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    TMhsw.close;
    TMhsw.Open;
    TMhsw.Refresh;
    FPRegisterTemplatel.Disconnect;
    DMFP.FPGetTemplatel.Run(1,perr);
end;
procedure TFEDMhs.SpeedButton2Click(Sender: TObject);
begin
    TMhsw.Prior;
end;
procedure TFEDMhs.SpeedButton3Click(Sender: TObject);
begin
    TMhsw.Next;
end;
procedure TFEDMhs.SpeedButton1Click(Sender: TObject);
begin
    if
dm.TMtKuliah.Locate('kodemtk',CBMakul.Text,[locaseinsensitive])
then
    begin
        dm.TMakulMhs.Open;
        if
not(dm.TMakulMhs.Locate('idmhs;kdmakul',vararrayof([TMhswnoid.Va
lue,CBMakul.Text]),[locaseinsensitive])) then
            begin
                dm.TMakulMhs.Insert;
                Dm.TMakulMhsidmhs.Value:=TMhswnoid.Value;
                Dm.TMakulMhsnmhs.Value:=TMhswnama.Value;
                Dm.TMakulMhskdmakul.Value:=Dm.TMtKuliahkodemtk.Value;
                Dm.TMakulMhsnmakul.Value:=Dm.TMtKuliahnama.Value;
                dm.TMakulMhs.Post;
            end
            else
            begin
                ShowMessage('Mahasiswa '+TMhswnama.Value+' Sudah
Mengambil Mata Kuliah '+Dm.TMtKuliahnama.Value);
            end; end; end;
procedure TFEDMhs.CBMakulChange(Sender: TObject);
begin
//if TMhsw.state in [dsedit,dsinsert] then
    begin

```

```

    getid.Close;
    getid.SQL.Clear;
    getid.SQL.Add('select * from tmakuldsn where
kdmakul=:kdmakul');
    getid.ParamByName('kdmakul').Value:=CBMakul.Text;
    getid.Open;
    CBIdDosen.Items.Clear;
    CBIdDosen.Text:='';
    while not(getid.Eof) do
    begin
        CBIdDosen.Items.Add(getid.FieldValues['iddosen']);
        getid.Next;
    end; end; end;
procedure TFEDMhs.SpeedButton4Click(Sender: TObject);
begin
    if dm.TDosen.Locate('noid',CBIdDosen.Text,[locaseinsensitive])
then
    begin
        dm.TMakulMhs.Open;
        //if
not(dm.TMakulMhs.Locate('idmhs',TMhswnoid.Value,[locaseinsensiti
ve])) then
        if
not(dm.TMakulMhs.Locate('idmhs;kdmakul',vararrayof([TMhswnoid.Va
lue,CBMakul.Text]),[locaseinsensitive])) then
            begin
                //if DM.TMakulMhskdmakul.Value='' then
                //begin
                    ShowMessage('Mahasiswa Belum Memilih MataKuliah !!!'
);
                //end;
                {dm.TMakulMhs.Insert;
                Dm.TMakulMhsidmhs.Value:=TMhswnoid.Value;
                Dm.TMakulMhsnmmhs.Value:=TMhswnama.Value;
                Dm.TMakulMhsiddosen.Value:=Dm.TDosennoid.Value;
                Dm.TMakulMhsnmdosen.Value:=Dm.TDosennama.Value;
                dm.TMakulMhs.Post;}
            end
        else
        begin
            dm.TMakulMhs.Edit;
            Dm.TMakulMhsidmhs.Value:=TMhswnoid.Value;
            Dm.TMakulMhsnmmhs.Value:=TMhswnama.Value;
            Dm.TMakulMhsiddosen.Value:=Dm.TDosennoid.Value;
            Dm.TMakulMhsnmdosen.Value:=Dm.TDosennama.Value;
            dm.TMakulMhs.Post;
        end; end; end;

```

```

procedure TFEDMhs.SpeedButton5Click(Sender: TObject);
begin
IF not(MakulMhs.IsEmpty) then
begin
MakulMhs.Delete;
end; end;
procedure TFEDMhs.BitBtn2Click(Sender: TObject);
begin
DBImage1Db1Click(nil);
end; end.

// EDIT JADWAL

unit UFDJdwAjar;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdCtrls, DBCtrls, Mask, ExtCtrls, Grids, DBGrids,
Buttons, db,
ComCtrls, mySQLDbTables, sqltimst,dateutils;
type
TFDJadwal = class(TForm)
DBGrid1: TDBGrid;
Panel1: TPanel;
Label1: TLabel;
Label2: TLabel;
Label9: TLabel;
Label10: TLabel;
Label3: TLabel;
Label5: TLabel;
Label7: TLabel;
ETgl: TDateTimePicker;
EJam: TDateTimePicker;
TDosen: TmySQLTable;
TMhSiswa: TmySQLTable;
TMaKul: TmySQLTable;
TJadwalKul: TmySQLTable;
DSJadwalKul: TDataSource;
DSMaKul: TDataSource;
DSDosen: TDataSource;
DSMhSiswa: TDataSource;
TJadwalKulkdmtk: TStringField;
TJadwalKulNmMTK: TStringField;
TJadwalKulkelas: TStringField;
TJadwalKuljumjam: TIntegerField;
TJadwalKulnmdosen: TStringField;

```

```
TJadwalKulnip: TStringField;
TJadwalKulruang: TStringField;
TJadwalKulstatus: TStringField;
TJadwalKulnim: TStringField;
TJadwalKulnmmhsiswa: TStringField;
TMaKulkodemtk: TStringField;
TMaKulnama: TStringField;
TDosenoid: TStringField;
TDosennama: TStringField;
TDosenfoto: TBlobField;
TDosenjnskelamin: TStringField;
TDosentempatlahir: TStringField;
TDosentgllahir: TDateField;
TDosenagama: TStringField;
TDosenalamat: TStringField;
TDosenstatus: TStringField;
TDosenket: TStringField;
TDosenSidikjari: TBlobField;
TMhSiswanoid: TStringField;
TMhSiswanama: TStringField;
TMhSiswafoto: TBlobField;
TMhSiswasidikjari: TBlobField;
TMhSiswajnskelamin: TStringField;
TMhSiswatempatlahir: TStringField;
TMhSiswatgllahir: TDateField;
TMhSiswaagama: TStringField;
TMhSiswaalamat: TStringField;
TMhSiswaket: TStringField;
QJadwlKul: TmySQLQuery;
DataSourcel: TDataSource;
TJadwalKultgl: TDateField;
TJadwalKuljammasuk: TTimeField;
TJadwalKuljamseslesai: TTimeField;
Ejam2: TDateTimePicker;
Label12: TLabel;
getid: TmySQLQuery;
dsgetid: TDataSource;
tmhsmkdsn: TmySQLQuery;
dsmhskddsn: TDataSource;
DBGrid2: TDBGrid;
dmhsikut: TDataSource;
DBGrid3: TDBGrid;
tmhsmkdsnidmhs: TStringField;
tmhsmkdsnnmhs: TStringField;
tmhsmkdsnkdmakul: TStringField;
tmhsmkdsnnmmakul: TStringField;
tmhsmkdsniddosen: TStringField;
```

```

tmhsmkdsnnmdosen: TStringField;
EKdMakul: TComboBox;
EMakul: TEdit;
EKDdosen: TComboBox;
ENmDosen: TEdit;
ERuang: TComboBox;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
Badd: TSpeedButton;
BEdit: TSpeedButton;
Bdel: TSpeedButton;
SpeedButton6: TSpeedButton;
BSave: TSpeedButton;
BitBtn1: TBitBtn;
TMhsIkut: TMySQLQuery;
Bevel1: TBevel;
Label4: TLabel;
Label6: TLabel;
EnmMhs: TEdit;
ENIM: TEdit;
DBGrid4: TDBGrid;
getididdosen: TStringField;
getidnmdosen: TStringField;
getidkdmakul: TStringField;
getidnmmakul: TStringField;
Bevel2: TBevel;
DBGrid5: TDBGrid;
DataSource2: TDataSource;
procedure BEditClick(Sender: TObject);
procedure BaddClick(Sender: TObject);
procedure BdelClick(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure BSaveClick(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure TJadwalKulAfterScroll(DataSet: TDataSet);
    procedure Button1Click(Sender: TObject);
    procedure EKDDosenExit(Sender: TObject);
    procedure EKDMhsiswaExit(Sender: TObject);
    procedure EKDDosenChange(Sender: TObject);
    procedure EKdMakulChange(Sender: TObject);
private
    { Private declarations }

```

```

    public
        { Public declarations }
    end;
var
    FDJadwal: TFDJadwal;
implementation
uses DMUnit;
{$R *.dfm}
procedure TFDJadwal.BEditClick(Sender: TObject);
begin
// CBKdMakul.SetFocus;
    BEdit.Tag:=1;
    TJadwalKul.Edit;
end;
procedure TFDJadwal.BaddClick(Sender: TObject);
begin
//CBKdMakul.SetFocus;
ETgl.Date:=date;
EJam.Time:=time;
EKdMakul.Text:='';
EMakul.Text:='';
EKDDosen.Text:='';
ENmDosen.Text:='';
ERuang.Text:='';
tmhsmkdsn.close;
    tmhsmkdsn.SQL.Clear;
    tmhsmkdsn.SQL.Add('select * from tmakulmhs where
(kdmakul=:kdmakul) and (iddosen=:iddosen)');
    tmhsmkdsn.ParamByName('kdmakul').Value:=EKdMaKul.Text;
    tmhsmkdsn.ParamByName('iddosen').Value:=EKDDosen.Text;
    tmhsmkdsn.ExecSQL;
    tmhsmkdsn.Open;
TJadwalKul.Insert;
end;
procedure TFDJadwal.BdelClick(Sender: TObject);
begin
if TJadwalKul.State in [dsbrowse] then
    if TJadwalKul.IsEmpty=false then
        TJadwalKul.Delete;
end;
procedure TFDJadwal.SpeedButton6Click(Sender: TObject);
begin
    BEdit.Tag:=0;
    TJadwalKul.Cancel;
    EKdMakul.Text:='';
    EMakul.Text:='';
    EKDDosen.Text:='';

```



```

ENmDosen.Text:='';
ERuang.Text:='';
tmhsmkdsn.close;
    tmhsmkdsn.SQL.Clear;
    tmhsmkdsn.SQL.Add('select * from tmakulmhs where
(kdmakul=:kdmakul) and (iddosen=:iddosen)');
    tmhsmkdsn.ParamByName('kdmakul').Value:=EKdMaKul.Text;
    tmhsmkdsn.ParamByName('iddosen').Value:=EKDDosen.Text;
    tmhsmkdsn.ExecSQL;
    tmhsmkdsn.Open;
end;
procedure TFDJadwal.BSaveClick(Sender: TObject);
begin
if BEdit.tag=0 then
begin
    QJadwlKul.close;
    QJadwlKul.SQL.Clear;
    QJadwlKul.SQL.Append('select * from jadwlkuliah where
(jammasuk<=:jamawl) and (jamselesai>=:jamend) and (tgl=:tgl) and
(ruang=:ruang)');
    QJadwlKul.ParamByName('jamawl').Value:=TimeToStr(EJam.Time);
    QJadwlKul.ParamByName('jamend').Value:=TimeToStr(EJam2.Time);

QJadwlKul.ParamByName('tgl').Value:=FormatDateTime('yyyy/mm/dd',
ETgl.Date);
    QJadwlKul.ParamByName('ruang').Value:=ERuang.Text;
    QJadwlKul.ExecSQL;
    QJadwlKul.open;
    if QJadwlKul.RecordCount=0 then
    begin
        tmhsmkdsn.First;
        while not(tmhsmkdsn.Eof) do
        begin
            TJadwalKul.Insert;
            TJadwalKultgl.Value:=ETgl.Date;
            TJadwalKuljammasuk.Value:=EJam.Time;
            TJadwalKulnim.Value:=tmhsmkdsnidmhs.Value;
            TJadwalKulnmmhsiswa.Value:=tmhsmkdsnnmmhs.Value;
            TJadwalKuljamselesai.Value:=EJam2.Time;
            TJadwalKulkdmtk.Value:=EKdMaKul.Text;
            TJadwalKulNmMTK.Value:=EMaKul.Text;
            TJadwalKulnm Dosen.Value:=ENmDosen.Text;
            TJadwalKulnip.Value:=EKDDosen.Text;
            TJadwalKulruang.Value:=ERuang.Text;
            TJadwalKul.Post;
            tmhsmkdsn.Next;
        end; end

```

```

else
begin
    ShowMessage('Ruang '+ERuang.Text+' Sedang dipakai');
end; end
else
begin
    TJadwalKultgl.Value:=ETgl.Date;
    TJadwalKuljammasuk.Value:=EJam.Time;
    TJadwalKulnim.Value:=tmhsmkdsnidmhs.Value;
    TJadwalKulnmmhsiswa.Value:=tmhsmkdsnnmmhs.Value;
    TJadwalKuljamselesai.Value:=EJam2.Time;
    TJadwalKulkdmtk.Value:=EKdMaKul.Text;
    TJadwalKulNmMTK.Value:=EMaKul.Text;
    TJadwalKulnmdosen.Value:=ENmDosen.Text;
    TJadwalKulnip.Value:=EKDDosen.Text;
    TJadwalKulruang.Value:=ERuang.Text;
    TJadwalKulnim.Value:=ENIM.Text;
    TJadwalKulnmmhsiswa.Value:=EnmMhs.Text;
    TJadwalKul.Post;
end;
    EKdMakul.Text:='';
    EMakul.Text:='';
    EKDDosen.Text:='';
    ENmDosen.Text:='';
    ERuang.Text:='';
    tmhsmkdsn.close;
    tmhsmkdsn.SQL.Clear;
    tmhsmkdsn.SQL.Add('select * from tmakulmhs where
(kdmakul=:kdmakul) and (iddosen=:iddosen)');
    tmhsmkdsn.ParamByName('kdmakul').Value:=EKdMaKul.Text;
    tmhsmkdsn.ParamByName('iddosen').Value:=EKDDosen.Text;
    tmhsmkdsn.ExecSQL;
    tmhsmkdsn.Open;
    BEdit.Tag:=0;
end;
procedure TFDJadwal.BitBtn1Click(Sender: TObject);
begin
    CLOSE;
end;
procedure TFDJadwal.FormShow(Sender: TObject);
begin
    EKdMakul.Items.Clear;
    dm.TMtKuliah.Open;
    dm.TMtKuliah.First;
    while not(dm.TMtKuliah.Eof) do
    begin
        EKdMakul.Items.Append(dm.TMtKuliahkodemtk.Value);
    end;
end;

```

```

        //CBNmMakul.Items.Append(dm.TMtKuliahnama.Value);
        dm.TMtKuliah.Next;
    end;
{   CBNmDosen.Items.Clear;
    dm.TDosen.Open;
    dm.TDosen.First;
    while not(dm.TDosen.Eof) do
    begin
        CBKodeDosen.Items.Append(dm.TDosenoid.Value);
        CBNmDosen.Items.Append(dm.TDosennama.Value);
        dm.TDosen.Next;
    end;}
    TDosen.Open;
    TMhSiswa.Open;
    TMaKul.Open;
    TJadwalKul.Open;
    ERuang.Items.LoadFromFile('.\nmruang.txt');
    ETgl.Date:=Date;
    EJam.Time:=time;
    Ejam2.time:=inchour(time,1);
    EKdMakul.Text:='';
    EMakul.Text:='';
    EKDdosen.Text:='';
    ENmDosen.Text:='';
    ERuang.Text:='';
end;
procedure TFDJadwal.SpeedButton2Click(Sender: TObject);
begin
    TJadwalKul.Prior;
end;
procedure TFDJadwal.SpeedButton3Click(Sender: TObject);
begin
    TJadwalKul.Next;
end;
procedure TFDJadwal.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    TDosen.Close;
    TMhSiswa.Close;
    TMaKul.Close;
    TJadwalKul.Close;
end;
procedure TFDJadwal.TJadwalKulAfterScroll(DataSet: TDataSet);
begin
    if TJadwalKul.State in [dsbrowse] then
    begin
        ETgl.Date:=TJadwalKultgl.Value;
    end;
end;

```

```

    EJam.Time:=TJadwalKuljammasuk.Value;
    EJam2.Time:=TJadwalKuljamselesai.Value;
    EKdMakul.Text:=TJadwalKulkdmtk.Value;
    EMakul.Text:=TJadwalKulNmMTK.Value;
    EKDDosen.Text:=TJadwalKulnip.Value;
    ENmDosen.Text:=TJadwalKulnmdosen.Value;
    ERuang.Text:=TJadwalKulruang.Value;
    ENIM.Text:=TJadwalKulnim.Value;
    EnmMhs.Text:=TJadwalKulnmmhsiswa.Value;
end;
if TJadwalKul.IsEmpty then
begin
    ETgl.Date:=TJadwalKultgl.Value;
    EJam.Time:=TJadwalKuljammasuk.Value;
    EJam2.Time:=TJadwalKuljamselesai.Value;
end; end;
procedure TFDJadwal.Button1Click(Sender: TObject);
var datasql : string;
begin
datasql:=FormatDateTime('hh:nn:ss',EJam.Time);
{ mySQLQuery1.Close;
mySQLQuery1.SQL.Clear;
mySQLQuery1.SQL.Add('select * from jadwlkuliah where
(jammasuk<=:jam) and (jamselesai>=:jam) and (tgl=:tgl) and
(nip=:nip) and (nim=:nim)');
mySQLQuery1.ParamByName('tgl').Value:=FormatDateTime('yyyy-mm-
dd',ETgl.Date);
mySQLQuery1.ParamByName('jam').Value:=TimeToStr(EJam2.time);
mySQLQuery1.ParamByName('nip').Value:=Edit1.Text;
mySQLQuery1.ParamByName('nim').Value:=Edit2.Text;
mySQLQuery1.Open;} end;
procedure TFDJadwal.EKDDosenExit(Sender: TObject);
begin
{ if TJadwalKul.state in [dsedit,dsinsert] then
begin
    if TDosen.Locate('noid',EKDDosen.Text,[locaseinsensitive])
then
        TJadwalKulnmdosen.Value:=TDosennama.Value;
end;} end;
procedure TFDJadwal.EKDMhsiswaExit(Sender: TObject);
begin
{if TJadwalKul.state in [dsedit,dsinsert] then
begin
    if
TMhSiswa.Locate('noid',EKDMhsiswa.Text,[locaseinsensitive]) then
        TJadwalKulnmmhsiswa.Value:=TMhSiswanama.Value;
end;} end;

```

```

procedure TFDJadwal.EKDDosenChange(Sender: TObject);
begin
// if TJadwalKul.state in [dsedit,dsinsert] then
// begin
    if TDosen.Locate('noid',EKDDosen.Text,[locaseinsensitive])
then
        ENmDosen.Text:=TDosennama.Value;
        //select * from tmakulmhs where (kdmakul=:kdmakul) and
(iddosen=:iddosen)
        tmhsmkdsn.close;
        tmhsmkdsn.SQL.Clear;
        tmhsmkdsn.SQL.Add('select * from tmakulmhs where
(kdmakul=:kdmakul) and (iddosen=:iddosen)');
        tmhsmkdsn.ParamByName('kdmakul').Value:=EKdMaKul.Text;
        tmhsmkdsn.ParamByName('iddosen').Value:=EKDDosen.Text;
        tmhsmkdsn.ExecSQL;
        tmhsmkdsn.Open;
        TMhsIkut.Close;
        TMhsIkut.Open;
//end; end;
procedure TFDJadwal.EKdMakulChange(Sender: TObject);
begin
    TMaKul.Locate('kodemtk',EKdMakul.Text,[locaseinsensitive]);
    EMakul.Text:=TMaKulnama.Value;
    getid.Close;
    getid.SQL.Clear;
    getid.SQL.Add('select * from tmakuldsn where
kdmakul=:kdmakul');
    getid.ParamByName('kdmakul').Value:=EKdMaKul.Text;
    getid.Open;
    EKDDosen.Items.Clear;
    while not(getid.Eof) do
    begin
        EKDDosen.Items.Add(getid.FieldValues['iddosen']);
        getid.Next;
    end; end; end.

// PREVIEW MATAKULIAH

unit Utmtkuliah;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, DB, mySQLDbTables, StdCtrls, Buttons, Grids, DBGrids,
ExtCtrls;
type

```

```

TFMtKuliah = class(TForm)
  Panell1: TPanel;
  GroupBox1: TGroupBox;
  Label1: TLabel;
  Enm: TEdit;
  Panel2: TPanel;
  DBGrid1: TDBGrid;
  Panel4: TPanel;
  BitBtn1: TBitBtn;
  dsmtkul: TDataSource;
  Tmtkul: TmySQLTable;
  Tmtkulkodemtk: TStringField;
  Tmtkulnama: TStringField;
  Tmtkulsks: TIntegerField;
  procedure BitBtn1Click(Sender: TObject);
  procedure FormHide(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure EnmKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  FMtKuliah: TFMtKuliah;
implementation
{$R *.dfm}
procedure TFMtKuliah.BitBtn1Click(Sender: TObject);
begin
  close;
end;
procedure TFMtKuliah.FormHide(Sender: TObject);
begin
  Tmtkul.Close;
end;
procedure TFMtKuliah.FormShow(Sender: TObject);
begin
  Tmtkul.Open;
end;
procedure TFMtKuliah.EnmKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  Tmtkul.Locate('nama',Enm.Text,[lopartialkey]);
end; end.

// PRIVIEW DATA DOSEN

```

```

unit UTDosen;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, DB, mySQLDbTables, Grids, DBGrids, ExtCtrls,
  StdCtrls, Buttons;

type
  TFtdosen = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel4: TPanel;
    DBGrid1: TDBGrid;
    TDosen: TmySQLTable;
    DSDosen: TDataSource;
    TDosennoId: TStringField;
    TDosennama: TStringField;
    TDosenfoto: TBlobField;
    TDosenjnskelamin: TStringField;
    TDosentempatlahir: TStringField;
    TDosentgllahir: TDateField;
    TDosenagama: TStringField;
    TDosenalamat: TStringField;
    TDosenstatus: TStringField;
    TDosenket: TStringField;
    TDosenSidikjari: TBlobField;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    ENmDosen: TEdit;
    EAlmt: TEdit;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormHide(Sender: TObject);
    procedure ENmDosenKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure EAlmtKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var

```

```

    Ftdosen: TFtdosen;
implementation
uses DMUnit;
{$R *.dfm}
procedure TFtdosen.BitBtn1Click(Sender: TObject);
begin
    close;
end;
procedure TFtdosen.FormShow(Sender: TObject);
begin
    TDosen.Open;
end;
procedure TFtdosen.FormHide(Sender: TObject);
begin
    TDosen.Close;
end;
procedure TFtdosen.ENmDosenKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    TDosen.Locate('nama',ENmDosen.Text,[lopartialkey]);
end;
procedure TFtdosen.EAlmtKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    TDosen.Locate('alamat',EAlmt.Text,[lopartialkey]);
end; end.

```

```
// PRIVIEW DATA MAHASISWA
```

```

unit UTMHsiswa;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, DB, mySQLDbTables, StdCtrls, Buttons, Grids, DBGrids,
    ExtCtrls;
type
    TFMhsiswa = class(TForm)
        Panel1: TPanel;
        GroupBox1: TGroupBox;
        Label1: TLabel;
        Label2: TLabel;
        Enm: TEdit;
        Ealmt: TEdit;
        Panel2: TPanel;
        DBGrid1: TDBGrid;
        Panel4: TPanel;
    end;

```



```

    BitBtn1: TBitBtn;
    DSmhsiswa: TDataSource;
    Tmhsiswa: TmySQLTable;
    Tmhsiswanoid: TStringField;
    Tmhsiswanama: TStringField;
    Tmhsiswafoto: TBlobField;
    Tmhsiswasidikjari: TBlobField;
    Tmhsiswajnskelamin: TStringField;
    Tmhsiswatempatlahir: TStringField;
    Tmhsiswatgllahir: TDateField;
    Tmhsiswaagama: TStringField;
    Tmhsiswaaalamat: TStringField;
    Tmhsiswaket: TStringField;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormHide(Sender: TObject);
    procedure EnmKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
    procedure EalmtKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    FMhsiswa: TFMhsiswa;
implementation
    {$R *.dfm}
    procedure TFMhsiswa.BitBtn1Click(Sender: TObject);
    begin
    CLOSE;
    end;
    procedure TFMhsiswa.FormShow(Sender: TObject);
    begin
    Tmhsiswa.Open;
    end;
    procedure TFMhsiswa.FormHide(Sender: TObject);
    begin
    Tmhsiswa.Close;
    end;
    procedure TFMhsiswa.EnmKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
    begin
    Tmhsiswa.Locate('nama',Enm.Text,[lopartialkey]);
    end;
    procedure TFMhsiswa.EalmtKeyUp(Sender: TObject; var Key: Word;

```

```

    Shift: TShiftState);
begin
Tmhsiswa.Locate('alamat',Ealmt.Text,[lopartialkey]);
end; end.

// PRIVIEW DATA JADWAL

unit UTJADWAL;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, DB, mySQLDbTables, StdCtrls, Buttons, Grids, DBGrids,
ExtCtrls;
type
    TFTJadwal = class(TForm)
        Panel1: TPanel;
        GroupBox1: TGroupBox;
        Label1: TLabel;
        ENm1: TEdit;
        ENm2: TEdit;
        Panel2: TPanel;
        Panel4: TPanel;
        BitBtn1: TBitBtn;
        DSJadwal: TDataSource;
        TJadwal: TmySQLTable;
        TJadwalkdmtk: TStringField;
        TJadwalNmMTK: TStringField;
        TJadwalkelas: TStringField;
        TJadwaltgl: TDateField;
        TJadwaljammasuk: TTimeField;
        TJadwaljamselesai: TTimeField;
        TJadwaljumjam: TIntegerField;
        TJadwalnmdosen: TStringField;
        TJadwalnip: TStringField;
        TJadwalruang: TStringField;
        TJadwalstatus: TStringField;
        TJadwalnim: TStringField;
        TJadwalnmmhsiswa: TStringField;
        Label2: TLabel;
        DBGrid2: TDBGrid;
        DBGrid3: TDBGrid;
        DBGrid1: TDBGrid;
        Label3: TLabel;
        Label4: TLabel;
        QMhsiswaMaKul: TmySQLQuery;
        QDosenMaKul: TmySQLQuery;
    end;
end;

```

```
DSDsnMakul: TDataSource;
DSMhsMakul: TDataSource;
CBIdDsn: TComboBox;
QDosenMaKulnoid: TStringField;
QDosenMaKulnama: TStringField;
QDosenMaKulfoto: TBlobField;
QDosenMaKuljnskelamin: TStringField;
QDosenMaKultempatlahir: TStringField;
QDosenMaKultgllahir: TDateField;
QDosenMaKulagama: TStringField;
QDosenMaKulalamat: TStringField;
QDosenMaKulstatus: TStringField;
QDosenMaKulket: TStringField;
QDosenMaKulSidikjari: TBlobField;
QDosenMaKuliddosen: TStringField;
QDosenMaKulnmdosen: TStringField;
QDosenMaKulkdmakul: TStringField;
QDosenMaKulnmmakul: TStringField;
QMhsiswaMaKulnoid: TStringField;
QMhsiswaMaKulnama: TStringField;
QMhsiswaMaKuljnskelamin: TStringField;
QMhsiswaMaKultempatlahir: TStringField;
QMhsiswaMaKultgllahir: TDateField;
QMhsiswaMaKulagama: TStringField;
QMhsiswaMaKulalamat: TStringField;
QMhsiswaMaKulket: TStringField;
QMhsiswaMaKulidmhs: TStringField;
QMhsiswaMaKulnmmhs: TStringField;
QMhsiswaMaKulkdmakul: TStringField;
QMhsiswaMaKulnmmakul: TStringField;
cbidmhs: TComboBox;
TMhsiswa: TmySQLTable;
DSMHsiswa: TDataSource;
TMhsiswanoid: TStringField;
TMhsiswanama: TStringField;
TMhsiswafoto: TBlobField;
TMhsiswasidikjari: TBlobField;
TMhsiswajnskelamin: TStringField;
TMhsiswatempatlahir: TStringField;
TMhsiswatgllahir: TDateField;
TMhsiswaagama: TStringField;
TMhsiswaalamat: TStringField;
TMhsiswaket: TStringField;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
```

```

    procedure FormHide(Sender: TObject);
    procedure ENm1KeyUp(Sender: TObject; var Key: Word;
        Shift: TShiftState);
    procedure ENm2KeyUp(Sender: TObject; var Key: Word;
        Shift: TShiftState);
    procedure CBIdDsnChange(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure cbidmhsChange(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    FTJadwal: TFTJadwal;
implementation
uses DMUnit, URepdosen, URepMhs;
{$R *.dfm}
procedure TFTJadwal.BitBtn1Click(Sender: TObject);
begin
    close;
end;
procedure TFTJadwal.FormShow(Sender: TObject);
begin
    TJadwal.Open;
    dm.TDosen.Open;
    CBIdDsn.Items.Clear;
    while not(Dm.TDosen.Eof) do
    begin
        CBIdDsn.Items.Add(Dm.TDosen.noid.Value);
        dm.TDosen.Next;
    end;

    TMhsiswa.Open;
    cbidmhs.Items.Clear;
    while not(TMhsiswa.Eof) do
    begin
        cbidmhs.Items.Add(TMhsiswanoid.Value);
        TMhsiswa.Next;
    end;
end;
procedure TFTJadwal.FormHide(Sender: TObject);
begin
    TJadwal.close;
end;
procedure TFTJadwal.ENm1KeyUp(Sender: TObject; var Key: Word;

```

```

    Shift: TShiftState);
begin
    TJadwal.Locate('nmdosen',Enm1.Text,[lopartialkey]);
end;
procedure TFTJadwal.ENm2KeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    TJadwal.Locate('nmmhsiswa',Enm2.Text,[lopartialkey]);
end;
procedure TFTJadwal.CBIdDsnChange(Sender: TObject);
begin
    QDosenMaKul.Close;
    QDosenMaKul.ParamByName('id').Value:=CBIdDsn.Text;
    QDosenMaKul.ExecSQL;
    QDosenMaKul.Open;
end;
procedure TFTJadwal.BitBtn2Click(Sender: TObject);
begin
    FRepDosen.repdosen.Preview;
end;
procedure TFTJadwal.cbidmhsChange(Sender: TObject);
begin
    QMHsiswaMaKul.Close;
    QMHsiswaMaKul.ParamByName('id').Value:=cbidmhs.Text;
    QMHsiswaMaKul.ExecSQL;
    QMHsiswaMaKul.Open;
end;
procedure TFTJadwal.BitBtn3Click(Sender: TObject);
begin
    FRepMhs.repmhs.Preview;
end; end.

// DATA MASUK RUANG

unit AbsU1;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, ToolWin, ActnMan, ActnCtrls, ActnMenus, StdActns,
    ActnList,
    ExtCtrls, UExtFP, StdCtrls, ComCtrls, Grids, DBGrids,
    dbtables, db,
    DBCtrls, Buttons, jpeg, mySQLDbTables,sqltimst,
    OleServer,
    DPSDKOPSLib_TLB,ComObj,OleCtrls,ActiveX,AXCtrls,DpSdkEngLib_TLB,
    DateUtils,

```

```
CPort;
type
TFrmMasuk = class(TForm)
    StatusB: TStatusBar;
    TQJadwal: TMySQLQuery;
    DSQJadwal: TDataSource;
    QCekDAbsen: TMySQLQuery;
    DSCekAbsen: TDataSource;
    TMhsw: TMySQLTable;
    DSMhsw: TDataSource;
    TJam: TTimer;
    FPGVerify: TFPGetTemplate;
    TVerify: TTimer;
    TMhswnoid: TStringField;
    TMhswnama: TStringField;
    TMhswfoto: TBlobField;
    TMhswsidikjari: TBlobField;
    TMhswjnskelamin: TStringField;
    TMhswtempatlahir: TStringField;
    TMhswtgllahir: TDateField;
    TMhswagama: TStringField;
    TMhswalamat: TStringField;
    TMhswket: TStringField;
    TDosen: TMySQLTable;
    DSDosen: TDataSource;
    TDosenoid: TStringField;
    TDosennama: TStringField;
    TDosenfoto: TBlobField;
    TDosenjnskelamin: TStringField;
    TDosentempatlahir: TStringField;
    TDosentgllahir: TDateField;
    TDosenagama: TStringField;
    TDosenalamat: TStringField;
    TDosenstatus: TStringField;
    TDosenket: TStringField;
    TDosenSidikjari: TBlobField;
    TDJdwlSkrng: TMySQLQuery;
    DSJdwlSkrng: TDataSource;
    TJadwalKul: TMySQLTable;
    TJadwalKulkdmtk: TStringField;
    TJadwalKulNmMTK: TStringField;
    TJadwalKulkelas: TStringField;
    TJadwalKuljumjam: TIntegerField;
    TJadwalKulnmdosen: TStringField;
    TJadwalKulnip: TStringField;
    TJadwalKulruang: TStringField;
    TJadwalKulstatus: TStringField;
```

```
TJadwalKulnim: TStringField;
TJadwalKulnmmhsiswa: TStringField;
TJadwalKultgl: TDateField;
TJadwalKuljammasuk: TTimeField;
TJadwalKuljamselesai: TTimeField;
DSJadwalKul: TDataSource;
TDJdwlSkrgkdmtk: TStringField;
TDJdwlSkrgNmMTK: TStringField;
TDJdwlSkrgkelas: TStringField;
TDJdwlSkrgtgl: TDateField;
TDJdwlSkrgjammasuk: TTimeField;
TDJdwlSkrgjamselesai: TTimeField;
TDJdwlSkrgjumjam: TIntegerField;
TDJdwlSkrgndosen: TStringField;
TDJdwlSkrgnip: TStringField;
TDJdwlSkrgruang: TStringField;
TDJdwlSkrgstatus: TStringField;
TDJdwlSkrgnim: TStringField;
TDJdwlSkrgnmmhsiswa: TStringField;
Panel2: TPanel;
DBGrid1: TDBGrid;
GroupBox2: TGroupBox;
Label2: TLabel;
LNim: TLabel;
LNama: TLabel;
Label1: TLabel;
Panel1: TPanel;
GbrFoto: TImage;
PJam: TPanel;
Panel5: TPanel;
PTglSkrg: TPanel;
PJamSkrg: TPanel;
Panel6: TPanel;
EJam: TDateTimePicker;
Panel3: TPanel;
Image1: TImage;
SpeedButton1: TSpeedButton;
GJadwal: TDBGrid;
TQJadwalkdmtk: TStringField;
TQJadwalNmMTK: TStringField;
TQJadwalkelas: TStringField;
TQJadwaltgl: TDateField;
TQJadwaljammasuk: TTimeField;
TQJadwaljamselesai: TTimeField;
TQJadwaljumjam: TIntegerField;
TQJadwalndosen: TStringField;
TQJadwalnip: TStringField;
```

```

TQJadwalruang: TStringField;
TQJadwalstatus: TStringField;
TQJadwalnim: TStringField;
TQJadwalnmhsw: TStringField;
DBGrid2: TDBGrid;
QCekDAbsentglmasuk: TDateField;
QCekDAbsenjammasuk: TTimeField;
QCekDAbsenkdmk: TStringField;
QCekDAbsenmmtk: TStringField;
QCekDAbsenkode: TStringField;
QCekDAbsennama: TStringField;
QCekDAbsenruang: TStringField;
QCekDAbsenstatus: TStringField;
DBGrid3: TDBGrid;
procedure ETDosenExecute(Sender: TObject);
procedure SetAboutExecute(Sender: TObject);
procedure EDtkarywnExecute(Sender: TObject);
procedure Action1Execute(Sender: TObject);
procedure Action3Execute(Sender: TObject);
procedure Action2Execute(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure DBGrid1Db1Click(Sender: TObject);
procedure TJamTimer(Sender: TObject);
procedure BStrStopClick(Sender: TObject);
procedure TQJadwalAfterScroll(DataSet: TDataSet);
procedure ETMhswExecute(Sender: TObject);
procedure ETMakulExecute(Sender: TObject);
procedure ETJadwalExecute(Sender: TObject);
procedure TVerifyTimer(Sender: TObject);
procedure FPGVerifyDone(Sender: TObject; var pTemplate:
OleVariant);
procedure MasukShow(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure UbhPasswExecute(Sender: TObject);
procedure MnExitExecute(Sender: TObject);
procedure Action4Execute(Sender: TObject);
procedure MnTDosenExecute(Sender: TObject);
procedure MnDMhswExecute(Sender: TObject);
procedure MnDMakulExecute(Sender: TObject);
procedure MnDJadwalExecute(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```



```

var
  FrmMasuk: TFrmMasuk;
  DSTEP : byte;
  TDatang,TJalan,dtserial,dttemp : string;
  hrini : array[1..7] of string =
('minggu','senin','selasa','Rabu','Kamis','Jumat','Sabtu');
  c_IVerify      : IFPVerify;
  c_regTemplate : IFPTemplate;
  endjam : word;
  endtm,wktstrt: tdatetime;
  strwktajar : string;
  perr : Aierrors;
implementation
uses ulogin, DMUnit, UEntryDK, UFDTMhsw, UFDMTK, UFDJdwAjar,
UScreenShot,
  UUbhPasswrD, UFAbsensi, UPASSW, UTAbsen, UTDosen, UTJADWAL,
UTMHsiswa,
  Utmtkuliah, UMForm;
{$R *.dfm}
procedure TFrmMasuk.ETDosenExecute(Sender: TObject);
begin
  if MessageDlg('Yakin Menghapus Semua data Dosen !
',mtConfirmation,[mbytes,mbcancel],0)=mryes then
  begin
    Dm.TDosen.Close;
    dm.TDosen.EmptyTable;
    dm.TDosen.Open;
  end
  else
    abort;
end;
procedure TFrmMasuk.SetAboutExecute(Sender: TObject);
begin
  FAbout.ShowModal;          //tampilkan form tentang pembuat
/ About
end;
procedure TFrmMasuk.EDtkarywnExecute(Sender: TObject);
begin
  FDosen.Show;              //tampilkan form entry data
mahasiswa
end;
procedure TFrmMasuk.Action1Execute(Sender: TObject);
begin
  FEDMhs.Show;
end;
procedure TFrmMasuk.Action3Execute(Sender: TObject);
begin

```

```

    FDMTK.Show;
end;
procedure TFrmMasuk.Action2Execute(Sender: TObject);
begin
    FDJadwal.Show;
end;
procedure TFrmMasuk.FormActivate(Sender: TObject);
begin
    TMhsw.Open;
end;
procedure TFrmMasuk.FormShow(Sender: TObject);
VAR PERR : AIERRORS;
begin
    TMhsw.Close;
    TMhsw.Open;
    TMhsw.Refresh;
    TDosen.Open;
    TJadwalKul.Open;
    Dm.TAbsensi.Open;
    DMFP.FPGetTemplatel.Run(1,pErr);
    TqJadwal.Close;
    TqJadwal.SQL.Clear;
    TqJadwal.SQL.Add('select * from jadwltkuliah where
(tgl=:tgl) ');
    //mySQLQuery1.SQL.Add('select * from jadwltkuliah where
(tgl=:tgl) and (nip=:nip) ');
    TqJadwal.ParamByName('tgl').Value:=FormatDateTime('yyyy-mm-
dd',Date);
    //TDJdwlSkrg.ParamByName('jam').Value:=TimeToStr(time);
    //TDJdwlSkrg.ParamByName('nip').Value:=noid;
    TqJadwal.Open;
end;
procedure TFrmMasuk.DBGrid1DbClick(Sender: TObject);
var jmstrt,jmskg : word;
begin
    { wktstrt:=mySQLQJdwlAjarjam.Value;
    jmstrt:=HourOf(mySQLQJdwlAjarjam.Value);
    endjam:=jmstrt+mySQLQJdwlAjarjumjam.Value;
    jmskg:=HourOf(time);
    if (jmskg>=jmstrt) and (jmskg<endjam) then
    begin
    endtm:=EncodeTime(endjam,00,00,00);
    PWAjar.Caption:='Waktu Kuliah Jam
'+TimeToStr(mySQLQJdwlAjarjam.Value) +' Sampai
'+TimeToStr(endtm);
    strwktajar:=PWAjar.Caption;
    PAbsen.Caption:='Silahkan Absen '};

```

```

    BStrStop.Enabled:=true;
    BStrStop.Tag:=1;
    Image1.Picture := nil;
    DMFP.FPGetTemplatel.Run(1,pErr);
end
else
begin
    ShowMessage('Jam Jadwal Mengajar Diluar Jam Sekarang !!!');
end;}end;
procedure TFrmMasuk.TJamTimer(Sender: TObject);
var mntsisa : word;
begin
    PJamSkrg.Caption:=TimeToStr(Time);
    PTglSkrg.Caption:=FormatDateTime('dd mmmm yyyy',date);
{ if BStrStop.Tag=1 then
begin
    mntsisa:=MinutesBetween(time,endtm);
    inc(mntsisa);
// PWAjar.Caption:=strwktajar+' Kurang dari :
'+IntToStr(mntsisa)+ ' menit';
    if CompareTime(endtm,time)<=0 then
        BStrStop.Click;
end;}end;
procedure TFrmMasuk.BStrStopClick(Sender: TObject);
VAR PERR : AIERRORS;
begin
{
    if BStrStop.tag=1 then
begin
        noid:='ds02';
        TVerify.Enabled:=False;
        BStrStop.Tag:=0;
        BStrStop.Caption:='Start';
        PAbsen.Caption:='-----';
        LNim.Caption:='-----';
        LNama.Caption:='-----';
        GbrFoto.Picture:=nil;
        //BStrStop.Enabled:=false;
        Image1.Picture := nil;
        dmfp.FPGetTemplatel.Disconnect;
        //DMFP.FPGetTemplatel.Disconnect(1,pErr);

end
else
begin
    noid:='ds02';
    TVerify.Enabled:=true;
    BStrStop.Tag:=1;

```

```

        BStrStop.Caption:='Stop';
        PAbsen.Caption:='-----';
        LNim.Caption:='-----';
        LNama.Caption:='-----';
        GbrFoto.Picture:=nil;
        //BStrStop.Enabled:=false;
        Image1.Picture := nil;
        //dmfp.FPGetTemplatel.Disconnect;
        DMFP.FPGetTemplatel.run(1,pErr);
    end;} end;
procedure TFrmMasuk.TQJadwalAfterScroll(DataSet: TDataSet);
begin
    {QMTKMHS.Close;
QMTKMHS.ParamByName('kdmakul').Value:=mySQLQJdwlAjarkdmtk.Value;
    QMTKMHS.ExecSQL;
    QMTKMHS.Open;}
end;
procedure TFrmMasuk.ETMhswExecute(Sender: TObject);
begin
    if MessageDlg('Yakin Menghapus Semua data Mahasiswa !
',mtConfirmation,[mbyes,mbcancel],0)=mryes then
    begin
        FrmMasuk.TMhsw.Close;
        FrmMasuk.TMhsw.EmptyTable;
        FrmMasuk.TMhsw.Open;
        DM.TMTKMHS.Close;
        DM.TMTKMHS.EmptyTable;
        DM.TMTKMHS.Open;
    end
    else
        abort;
end;
procedure TFrmMasuk.ETMakulExecute(Sender: TObject);
begin
    if MessageDlg('Yakin Menghapus Semua data Mata Kuliah !
',mtConfirmation,[mbyes,mbcancel],0)=mryes then
    begin
        Dm.TMtKuliah.Close;
        dm.TMtKuliah.EmptyTable;
        dm.TMtKuliah.Open;
    end
    else
        abort;
end;
procedure TFrmMasuk.ETJadwalExecute(Sender: TObject);
begin

```

```

if MessageDlg('Yakin Menghapus Semua data Jadwal Mengajar Dosen
! ',mtConfirmation,[mbyes,mbcancel],0)=mryes then
begin
    Dm.TJadwal.Close;
    dm.TJadwal.EmptyTable;
    dm.TJadwal.Open;
end
else
    abort;
end;
procedure TFrmMasuk.TVerifyTimer(Sender: TObject);
var picstream : tstream;
    dt : string;
    jmtlmbt,jmmasuk : TTime;
begin
TVerify.Enabled:=false;
if noid<>' ' then
begin
dt:=FormatDateTime('yyyy-mm-dd',Date);
dt:=TimeToStr(time);
if (pos('D',noid)>0) or (pos('d',noid)>0) then
begin
    picstream:=TDosen.CreateBlobStream(FrmMasuk.TDosenfoto,bmread);
    TDJdwlSkrng.Close;
    TDJdwlSkrng.SQL.Clear;
    TDJdwlSkrng.SQL.Add('select * from jadwلكuliah where
(jammasuk<=:jam) and (jamselesai>=:jam) and (tgl=:tgl) and
(nip=:nip) ');
    //mySQLQuery1.SQL.Add('select * from jadwلكuliah where
(tgl=:tgl) and (nip=:nip) ');
    TDJdwlSkrng.ParamByName('tgl').Value:=FormatDateTime('yyyy-mm-
dd',Date);
    TDJdwlSkrng.ParamByName('jam').Value:=TimeToStr(time);
    TDJdwlSkrng.ParamByName('nip').Value:=noid;
    TDJdwlSkrng.Open;
    LNim.Caption:=noid;
    LNama.Caption:=TDosennama.Value;
    if TDJdwlSkrng.RecordCount>0 then
begin
    jmmasuk:=TDJdwlSkrng.jammasuk.Value;
    jmtlmbt:=IncMinute(jmmasuk,15);
    jmmasuk:=Time;
    if CompareTime(jmmasuk,jmtlmbt)<=0 then
begin
        QCekDAbsen.Close;
        QCekDAbsen.SQL.Clear;

```

```

    QCekDAbsen.SQL.Add('select * from tabsenmhs where
(jammasuk<=:jam) and (tglmasuk=:tgl) and (kode=:nip)');
    //mySQLQuery1.SQL.Add('select * from jadwlkuliah where
(tgl=:tgl) and (nip=:nip)');
    QCekDAbsen.ParamByName('tgl').Value:=FormatDateTime('yyyy-
mm-dd',Date);
    QCekDAbsen.ParamByName('jam').Value:=TimeToStr(time);
    QCekDAbsen.ParamByName('nip').Value:=noid;
    QCekDAbsen.Open;
    if QCekDAbsen.RecordCount=0 then
    begin
    Dm.TAbsensi.Insert;
    Dm.TAbsensitglmasuk.Value:=date;
    Dm.TAbsensijammasuk.Value:=time;
    Dm.TAbsensikode.Value:=noid;
    Dm.TAbsensinama.Value:=TDJdwlSkrgnmdosen.Value;
    Dm.TAbsensikdmtk.Value:=TDJdwlSkrgkdmk.Value;
    Dm.TAbsensinmmtk.Value:=TDJdwlSkrgNmMTK.Value;
    Dm.TAbsensiruang.Value:=TDJdwlSkrgruang.Value;
    Dm.TAbsensistatus.Value:='Pengajar';
    Dm.TAbsensi.Post;
    end;
    dt:=copy(TDJdwlSkrgruang.Value,1,1);
    if dt='1' then
    begin
        if MForm.ComPort1.Connected then
            MForm.ComPort1.WriteString('BP1'+#13);
    end;
    if dt='2' then
    begin
        if MForm.ComPort1.Connected then
            MForm.ComPort1.WriteString('BP2'+#13);
    end;
    if dt='3' then
    begin
        if MForm.ComPort1.Connected then
            MForm.ComPort1.WriteString('BP3'+#13);
    end;
    end
    else
    begin
        ShowMessage('Anda Terlambat !!!');
    end; end; end
else
begin
    picstream:=TMhsw.CreateBlobStream(FrmMasuk.TMhswfoto,bmread);
    TDJdwlSkrg.Close;

```

```

TDJdwlSkrng.SQL.Clear;
TDJdwlSkrng.SQL.Add('select * from jadwlkuliah where
(jammasuk<=:jam) and (jamselesai>=:jam) and (tgl=:tgl) and
(nim=:nim)');
TDJdwlSkrng.ParamByName('tgl').Value:=FormatDateTime('yyyy-mm-
dd',Date);
TDJdwlSkrng.ParamByName('jam').Value:=TimeToStr(time);
TDJdwlSkrng.ParamByName('nim').Value:=noid;
TDJdwlSkrng.Open;
LNim.Caption:=noid;
LNama.Caption:=TMhswnama.Value;
if TDJdwlSkrng.RecordCount>0 then
begin
jmmasuk:=TDJdwlSkrngjammasuk.Value;
jmtlmbt:=IncMinute(jmmasuk,15);
jmmasuk:=Time;
if CompareTime(jmmasuk,jmtlmbt)<=0 then
begin
QCekDAbsen.Close;
QCekDAbsen.SQL.Clear;
QCekDAbsen.SQL.Add('select * from tabsenmhs where
(jammasuk<=:jam) and (tglmasuk=:tgl) and (kode=:nip)');
//mySQLQuery1.SQL.Add('select * from jadwlkuliah where
(tgl=:tgl) and (nip=:nip)');
QCekDAbsen.ParamByName('tgl').Value:=FormatDateTime('yyyy-
mm-dd',Date);
QCekDAbsen.ParamByName('jam').Value:=TimeToStr(time);
QCekDAbsen.ParamByName('nip').Value:=noid;
QCekDAbsen.Open;
if QCekDAbsen.RecordCount=0 then
begin
Dm.TAbsensi.Insert;
Dm.TAbsensitglmasuk.Value:=date;
Dm.TAbsensijammasuk.Value:=time;
Dm.TAbsensikode.Value:=noid;
Dm.TAbsensinama.Value:=TDJdwlSkrngmmhsiswa.Value;
Dm.TAbsensikdmtk.Value:=TDJdwlSkrngkdmtk.Value;
Dm.TAbsensinmmtk.Value:=TDJdwlSkrngNmMTK.Value;
Dm.TAbsensiruang.Value:=TDJdwlSkrngruang.Value;
Dm.TAbsensistatus.Value:='Mahasiswa';
Dm.TAbsensi.Post;
end;
dt:=copy(TDJdwlSkrngruang.Value,1,1);
if dt='1' then
begin
if MForm.ComPort1.Connected then
MForm.ComPort1.WriteStr('BP1'+#13);

```

```

end;

if dt='2' then
begin
    if MForm.ComPort1.Connected then
        MForm.ComPort1.WriteStr('BP2'+#13);
    end;
    if dt='3' then
    begin
        if MForm.ComPort1.Connected then
            MForm.ComPort1.WriteStr('BP3'+#13);
        end;end
    else
    begin
        ShowMessage('Anda Terlambat !!!');
    end;end;end;
GbrFoto.Picture.Bitmap.LoadFromStream(picstream);
picstream.Free;
end
else
begin
LNim.Caption:='Belum Registrasi';
LNama.Caption:='-----';
TDJdwlSkrg.Close;
GbrFoto.Picture:=nil;
end;
//if
not (TAbsenMhs.Locate('kode',LNim.Caption,[locaseinsensitive]))
then
begin
{  if QMTKMHS.Locate('kdmhs',LNim.Caption,[locaseinsensitive])
then
    begin
        TAbsenMhs.Insert;
        TAbsenMhstglmasuk.Value:=date;
        TAbsenMhsjammasuk.Value:=time;
        TAbsenMhskdmtk.Value:=mySQLQJdwlAjarkdmtk.Value;
        TAbsenMhsnmmtk.Value:=mySQLQJdwlAjarNmMTK.Value;
        TAbsenMhs.Post;
    end
    else
    begin
        ShowMessage('Mahasiswa Tidak Terdaftar Atau Belum Registrasi
!!!');
    end;}
end;
end;
{else

```



```

begin
    //ShowMessage('Mahasiswa Sudah Absen !!!');
end;} end;

procedure TFrmMasuk.FPGVerifyDone(Sender: TObject;
var pTemplate: OleVariant);
var
    l_verifyOk : WordBool;
    l_score : OleVariant;
    l_security : AISecureModeMask;
l_threshold : OleVariant;
    l_data : OleVariant;
    l_loop : Integer;
    l_learning : WordBool;

begin
    l_verifyOk:=false;
    while ( (not(FrmMasuk.TMhsw.Eof)) and (not(l_verifyOk=true))
) do
        begin
            l_verifyOk := False;
            l_data:=TMhswsidikjari.Value;
            c_RegTemplate.Import(l_data);

c_IVerify.Compare(c_RegTemplate,pTemplate,l_verifyOk,l_score,l_t
hreshold,l_learning,l_security);

            if l_verifyOk = False then
                begin
                    FrmMasuk.TMhsw.Next;
                end
            else
                begin
                    TVerify.Enabled:=true;
                    exit;
                end;
        end;
    end; // END WHILE

    ShowMessage('Mahasiswa Belum Registrasi Sidik Jari
!!!!');
end;
procedure TFrmMasuk.MasukShow(Sender: TObject);
begin
    TMhsw.close;
    TMhsw.Open;

```

```

TMhsw.Refresh;
end;
procedure TFrmMasuk.SpeedButton7Click(Sender: TObject);
begin
// RvProject1.Execute;
end;
procedure TFrmMasuk.UbhPasswExecute(Sender: TObject);
begin
    FUbhPassw.ShowModal;
end;
procedure TFrmMasuk.MnExitExecute(Sender: TObject);
begin
    Close;
end;
procedure TFrmMasuk.Action4Execute(Sender: TObject);
begin
    MForm.ComPort1.ShowSetupDialog;
    if MForm.ComPort1.Tag=1 then
    begin
        MForm.ComPort1.StoreSettings(stIniFile, '.\setcom.ini');
        MForm.ComPort1.Connected:=true;
    end; end;
procedure TFrmMasuk.MnTDosenExecute(Sender: TObject);
begin
    FtDosen.Show;
end;
procedure TFrmMasuk.MnDMhswiswExecute(Sender: TObject);
begin
    FMhswisw.Show;
end;
procedure TFrmMasuk.MnDMakulExecute(Sender: TObject);
begin
    FMtKuliah.Show;
end;
procedure TFrmMasuk.MnDJadwalExecute(Sender: TObject);
begin
    FTJadwal.Show;
end;
procedure TFrmMasuk.SpeedButton1Click(Sender: TObject);
begin
if GJadwal.tag=0 then
begin
    GJadwal.tag:=1;
    GJadwal.Visible:=true;
end
else
begin

```

```

    GJadwal.tag:=0;
    GJadwal.Visible:=false;
end;
end;
end.

// PREVIEW DATA ABSENSI

unit UTabsen;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, DB, mySQLDbTables, StdCtrls, Buttons, Grids, DBGrids,
    ExtCtrls;
type
    TFTAbsen = class(TForm)
        Panel1: TPanel;
        GroupBox1: TGroupBox;
        Label1: TLabel;
        Label2: TLabel;
        Enm: TEdit;
        Esta: TEdit;
        Panel2: TPanel;
        DBGrid1: TDBGrid;
        Panel4: TPanel;
        BitBtn1: TBitBtn;
        DSAbsen: TDataSource;
        TAbsen: TmySQLTable;
        TAbsentglmasuk: TDateField;
        TAbsenjammasuk: TTimeField;
        TAbsenkdmtdk: TStringField;
        TAbsennmtdk: TStringField;
        TAbsenkode: TStringField;
        TAbsennama: TStringField;
        TAbsenruang: TStringField;
        TAbsenstatus: TStringField;
        procedure EnmKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
        procedure EstaKeyUp(Sender: TObject; var Key: Word;
        Shift: TShiftState);
        procedure BitBtn1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
end;

```

```
var
  FTAbsen: TFTAbsen;
implementation
{$R *.dfm}
procedure TFTAbsen.EnmKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  TAbsen.Locate('nama',ENm.Text,[lopartialkey]);
end;
procedure TFTAbsen.EstaKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  TAbsen.Locate('status',ESta.Text,[lopartialkey]);
end;
procedure TFTAbsen.BitBtn1Click(Sender: TObject);
begin
  CLOSE;
end;end.
```