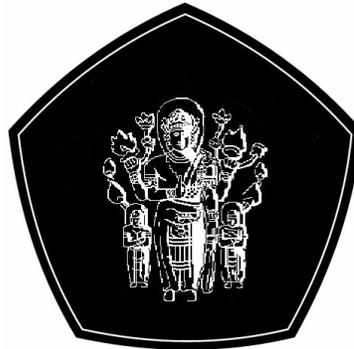


# **SISTEM INFORMASI KEHADIRAN MAHASISWA DENGAN MENGGUNAKAN BARCODE**

## **SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh :

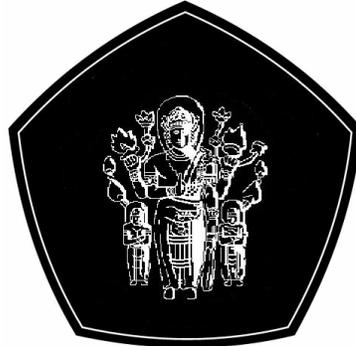
**SAID ISKANDAR**  
**NIM. 0001060373-63**

**DEPARTEMEN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
JURUSAN TEKNIK ELEKTRO  
MALANG  
2007**



# SISTEM INFORMASI KEHADIRAN MAHASISWA DENGAN MENGGUNAKAN BARCODE SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh :

**SAID ISKANDAR**  
**NIM. 0001060373-63**

Telah diperiksa dan disetujui oleh  
Dosen Pembimbing :

**Ir. Primantara HT.**  
**NIP. 132 090 390**

**Arief Andy Soebroto, ST., M.Kom.**  
**NIP. 132 231 567**



## SISTEM INFORMASI KEHADIRAN MAHASISWA DENGAN MENGGUNAKAN BARCODE

Disusun oleh :

**SAID ISKANDAR**  
**NIM. 0001060373-63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 07 Februari 2007

**Majelis Penguji :**

**Ir. Sutrisno**  
**NIP. 131 653 479**

**Ir. Heru Nurwasito, M.Kom.**  
**NIP. 131 879 033**

**Herman Tolle, ST., MT.**  
**NIP. 132 283 206**

**Ir. Muhammad Aswin**  
**NIP. 131 879 045**

Mengetahui :  
Ketua Jurusan Teknik Elektro

**Ir. Purwanto, MT.**  
**NIP. 131 574 847**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Atas rahmat Allah Yang Maha Penyayang , tugas akhir ini kupersembahkan untuk Papa dan Mama tercinta serta saudaraku atas doa dan kasih sayangnya. Terima kasih untuk teman-teman yang memberikan inspirasi kehidupan dan orang-orang di sekelilingku yang kusayang dan menyayangiku.



## KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah atas segala rahmat, nikmat serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Sistem Informasi Kehadiran Mahasiswa Dengan Menggunakan Barcode”. Penulis menyadari bahwa dalam penyelesaian tugas akhir ini banyak mendapat bantuan, dukungan, motivasi, dan fasilitas dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada :

1. Bapak Ir. Purwanto, MT dan Bapak Ir. Heri Purnomo selaku Ketua dan Sekretaris Jurusan Teknik Elektro dan segenap Bapak dan Ibu Dosen, Staf Administrasi dan Perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
2. Bapak Ir. Heru Nurwarsito, M.Kom selaku KKDK Sistem Informatika dan Komputer.
3. Bapak Ir. Primantara Hari Trisnawan selaku Dosen Pembimbing I yang telah banyak memberikan masukan dan arahan dalam penyusunan tugas akhir ini.
4. Bapak Arief Andy Soebroto, ST., M.Kom. selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan dan arahan terhadap penyusunan tugas akhir ini.
5. Kedua orangtuaku, Bapak Said Mahdar dan Ibunda Syarifah Fauziah untuk doa dan harapan yang selalu mengiringi dalam perjalanan penyelesaian tugas akhir ini.
6. Kakak-kakak ku, Syarifah Safia, Saïd Hanafi dan keluarga, Saïd Haris dan keluarga, untuk doa dan dukungan sehingga penulis dapat menyelesaikan tugas akhir.
7. Teman-teman TPTIFT yang memberikan tempat untuk pengerjaan tugas akhir dan ide-ide segar tentang kehidupan.
8. Adik yang mengingatkan penulis untuk selalu semangat dalam mengerjakan tugas akhir.
9. Teman-teman semasa SMU dan kuliah yang tidak dapat disebutkan satu-persatu, terima kasih atas pengalaman dan kenangan yang tidak terlupakan.
10. Semua pihak yang telah membantu penulis dalam menyelesaikan penyusunan tugas akhir ini.

Kesempurnaan hanya milik Allah semata, tiada daya dan upaya kecuali hanya dari-Nya. Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, untuk itu saran dan kritik yang bersifat membangun sangat diharapkan guna untuk perbaikan yang akan datang. Akhir kata, penulis berharap semoga tugas akhir ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, Januari 2007

Penulis

UNIVERSITAS BRAWIJAYA



## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	<b>i</b>
<b>DAFTAR ISI</b> .....	<b>iii</b>
<b>DAFTAR TABEL</b> .....	<b>x</b>
<b>DAFTAR GAMBAR</b> .....	<b>xi</b>
<b>DAFTAR ISTILAH</b> .....	<b>xix</b>
<b>ABSTRAK</b> .....	<b>xxi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	4
<b>BAB II TEORI DASAR</b> .....	<b>5</b>
2.1 Sistem Informasi .....	5
2.1.1 Siklus Informasi .....	5
2.1.2 Perangkat Pengadaan Sistem Informasi .....	7
2.1.2.1 <i>Data Flow Diagram Context Level</i> (Diagram Konteks) .....	7
2.1.2.2 <i>Data Flow Diagram Levelled</i> (DFD Levelled) .....	8
2.1.2.3 <i>State Transition Diagram</i> (STD) .....	9
2.2 Basis Data .....	10
2.2.1 Perancangan Basis Data .....	10
2.2.1.1 <i>Entity-Relationship Diagram</i> .....	10
2.2.1.2 Normalisasi .....	12
2.2.1.2.1 <i>Lossless-Join Decomposition</i> .....	13
2.2.1.2.2 <i>Dependency Preservation</i> .....	13
2.2.1.2.3 <i>Boyce-Code Normal Form</i> (BCNF) .....	13
2.2.1.2.4 Bentuk Normal Tahap Pertama ( <i>1<sup>st</sup> Normal Form</i> ) .....	14
2.2.1.2.5 Bentuk Normal Tahap Kedua ( <i>2<sup>st</sup> Normal Form</i> ) .....	14
2.2.1.2.6 Bentuk Normal Tahap Ketiga ( <i>3<sup>st</sup> Normal Form</i> ) .....	14
2.2.2 Sistem Manajemen Basis Data .....	14
2.2.2.1 Abstraksi Data .....	15
2.2.2.2 Model Data .....	15
2.2.2.2.1 Model Data Hirarki .....	15
2.2.2.2.2 Model Data Jaringan .....	16
2.2.2.2.3 Model Data Relasional .....	17

2.2.2.3	Arsitektur Sistem.....	18
2.2.2.3.1	Sistem Tunggal ( <i>Stand Alone</i> ).....	18
2.2.2.3.2	Sistem Terpusat ( <i>Centralized System</i> ).....	18
2.2.2.3.3	Sistem <i>Client Server</i> .....	19
2.2.3	<i>Structure Query Language</i> (SQL).....	20
2.2.3.1	<i>Data Definition Language</i> (DDL).....	21
2.2.3.2	<i>Data Manipulation Language</i> (DML).....	21
2.2.4	<i>Sybase Power Designer 10</i> .....	21
2.2.4.1	<i>Conceptual Data Model</i> (CDM).....	22
2.2.4.2	<i>Physical Data Model</i> (PDM).....	23
2.2.5	Borland Interbase.....	24
2.2.5.1	Hal Utama <i>Interbase Server</i> .....	24
2.2.5.2	Konsep <i>Client-Server</i> di Interbase.....	25
2.2.5.2.1	<i>Client</i> .....	25
2.2.5.2.2	<i>Server</i> .....	26
2.3	Borland Delphi.....	26
2.3.1	Lingkungan IDE ( <i>Integrated Development Environment</i> ) Delphi.....	28
2.4	Barcode.....	29
2.4.1	EAN ( <i>European Article Numbering</i> ).....	30
2.4.2	<i>Code 128</i> .....	31
2.4.3	<i>Code 93</i> .....	31
2.4.4	<i>Code 39</i> .....	32
2.4.5	UPC ( <i>Universal Product Code</i> ).....	32
2.5	Jaringan Komputer.....	34
2.5.1	Jenis Jaringan Komputer.....	34
2.5.2	Model Referensi TCP/IP ( <i>Transfer Control Protocol/Internet Protocol</i> ).....	35
2.5.3	Topologi Jaringan Komputer.....	37
2.5.4	Manfaat Jaringan Komputer.....	39
<b>BAB III METODE PENELITIAN .....</b>		<b>40</b>
3.1	Studi Literatur.....	40
3.2	Perancangan Perangkat Lunak.....	40
3.3	Implementasi Perangkat Lunak.....	41
3.4	Pengujian dan Analisis Perangkat Lunak.....	41
3.5	Pengambilan Kesimpulan dan Saran.....	41
<b>BAB IV PERANCANGAN PERANGKAT LUNAK.....</b>		<b>42</b>
4.1	Analisis Kebutuhan Perangkat Lunak.....	43
4.1.1	Analisis Administrasi Kehadiran Mahasiswa.....	44

4.1.2 Analisis Administrasi Kehadiran Mahasiswa Secara Manual.....	45
4.1.3 Analisis Sistem Informasi Kehadiran Mahasiswa.....	46
4.2 Perancangan Sistem .....	48
4.2.1 Diagram Blok Sistem .....	48
4.2.2 Diagram Konteks .....	49
4.2.3 Cara Kerja Sistem .....	50
4.3 Perancangan Basis Data .....	51
4.3.1 <i>Entity Relationship Diagram</i> (ERD).....	51
4.3.2 <i>Data Object Description</i> .....	52
4.3.2.1 <i>Data Type Definition</i> .....	52
4.3.2.2 <i>Data Field Description</i> .....	54
4.3.3 Desain Basis Data Kehadiran Ternormalisasi .....	56
4.4 Perancangan Proses.....	56
4.4.1 <i>Data Flow Diagram</i> (DFD) .....	57
4.4.1.1 <i>Data Flow Diagram Level 0 Aplikasi Server</i> .....	58
4.4.1.2 <i>Data Flow Diagram Level 0 Aplikasi Client</i> .....	58
4.4.1.3 <i>Data Flow Diagram Level 1 Aplikasi Server</i> .....	60
4.4.1.4 <i>Data Flow Diagram Level 2 Proses Administrasi Data Mahasiswa</i> .....	63
4.4.1.5 <i>Data Flow Diagram Level 2 Proses Administrasi Data Dosen</i> .....	64
4.4.1.6 <i>Data Flow Diagram Level 2 Proses Administrasi Data Mata Kuliah</i> .....	65
4.4.1.7 <i>Data Flow Diagram Level 2 Proses Administrasi Data Kurikulum</i> .....	66
4.4.1.8 <i>Data Flow Diagram Level 2 Proses Administrasi Data Login</i> .....	67
4.4.1.9 <i>Data Flow Diagram Level 2 Proses Administrasi Data Peserta Mata Kuliah</i> .....	68
4.4.1.10 <i>Data Flow Diagram Level 2 Proses Administrasi Data Jadwal Mata Kuliah</i> .....	69
4.4.1.11 <i>Data Flow Diagram Level 2 Proses Administrasi Data Kehadiran Mahasiswa</i> .....	70
4.4.2 <i>State Transition Diagram</i> (STD) Proses Pencatatan Kehadiran Mahasiswa .....	71
4.4.3 Diagram Alir Sistem Informasi Mahasiswa .....	72
4.4.3.1 Diagram Alir Aplikasi <i>Server</i> .....	73
4.4.3.1.1 Diagram Alir Proses <i>Login Server</i> .....	73
4.4.3.1.2 Diagram Alir Proses Administrasi Data Mahasiswa .....	74
4.4.3.1.3 Diagram Alir Proses Administrasi Data Dosen.....	76
4.4.3.1.4 Diagram Alir Proses Administrasi Data Mata Kuliah.....	78
4.4.3.1.5 Diagram Alir Proses Administrasi Data Kurikulum .....	80

4.4.3.1.6	Diagram Alir Proses Administrasi Data <i>Login</i> .....	82
4.4.3.1.7	Diagram Alir Proses Administrasi Data Peserta Mata Kuliah ..	83
4.4.3.1.8	Diagram Alir Proses Administrasi Data Jadwal Mata Kuliah..	85
4.4.3.1.9	Diagram Alir Proses Administrasi Data Kehadiran Mahasiswa.....	87
4.4.3.1.10	Diagram Alir Proses Pencetakan Kartu Kehadiran Mahasiswa (KKM).....	89
4.4.3.1.11	Diagram Alir Proses Pencetakan Kartu Kehadiran Dosen (KKD) .....	89
4.4.3.1.12	Diagram Alir Proses Pencetakan <i>Report</i> Kehadiran Mahasiswa.....	90
4.4.3.2	Diagram Alir Aplikasi <i>Client</i> .....	91
4.4.3.2.1	Diagram Alir Proses <i>Login User</i> .....	92
4.5	Perancangan Antarmuka Sistem Informasi Kehadiran Mahasiswa .....	94
4.5.1	Perancangan Antarmuka Aplikasi <i>Server</i> .....	95
4.5.1.1	Perancangan Antarmuka <i>Login Server</i> .....	96
4.5.1.2	Perancangan Antarmuka Administrasi Data Mahasiswa .....	97
4.5.1.3	Perancangan Antarmuka Administrasi Data Dosen.....	98
4.5.1.4	Perancangan Antarmuka Administrasi Data Mata kuliah.....	99
4.5.1.5	Perancangan Antarmuka Administrasi Data Kurikulum.....	100
4.5.1.6	Perancangan Antarmuka Administrasi Data <i>Login</i> .....	101
4.5.1.7	Perancangan Antarmuka Administrasi Data Peserta Mata Kuliah.....	102
4.5.1.8	Perancangan Antarmuka Administrasi Data Jadwal Mata Kuliah .....	103
4.5.1.8.1	Perancangan Antarmuka Administrasi Data Jadwal Mata Kuliah Tambahan	104
4.5.1.9	Perancangan Antarmuka Administrasi Data Kehadiran Mahasiswa.....	105
4.5.1.10	Perancangan Antarmuka Pencetakan Data Kartu Kehadiran Mahasiswa.....	106
4.5.1.11	Perancangan Antarmuka Pencetakan Data Kartu Kehadiran Dosen .....	106
4.5.1.12	Perancangan Antarmuka Pencetakan Data <i>Report</i> Kehadiran Mahasiswa.....	107
4.5.2	Perancangan Antarmuka Aplikasi <i>Client</i> .....	108
<b>BAB V IMPLEMENTASI PERANGKAT LUNAK.....</b>		<b>110</b>
5.1	Implementasi Perancangan Basis Data .....	112
5.2	Implementasi Antarmuka Aplikasi <i>Server</i> .....	114
5.2.1	Implementasi Perangkat Lunak untuk <i>Login Server</i> .....	115
5.2.2	Implementasi Perangkat Lunak untuk Administrasi Data Mahasiswa.....	117

5.2.2.1 Implementasi Penambahan Data Mahasiswa .....	117
5.2.2.2 Implementasi Pengubahan Data Mahasiswa .....	118
5.2.2.3 Implementasi Penghapusan Data Mahasiswa .....	119
5.2.2.4 Implementasi Pencarian Data Mahasiswa .....	119
5.2.3 Implementasi Perangkat Lunak untuk Administrasi Data Dosen .....	120
5.2.3.1 Implementasi Penambahan Data Dosen .....	120
5.2.3.2 Implementasi Pengubahan Data Dosen .....	121
5.2.3.3 Implementasi Penghapusan Data Dosen .....	122
5.2.3.4 Implementasi Pencarian Data Dosen .....	123
5.2.4 Implementasi Perangkat Lunak untuk Administrasi Data Mata Kuliah .....	124
5.2.4.1 Implementasi Penambahan Data Mata Kuliah .....	124
5.2.4.2 Implementasi Pengubahan Data Mata Kuliah .....	125
5.2.4.3 Implementasi Penghapusan Data Mata Kuliah .....	126
5.2.4.4 Implementasi Pencarian Data Mata Kuliah .....	127
5.2.5 Implementasi Perangkat Lunak untuk Administrasi Data Kurikulum .....	128
5.2.5.1 Implementasi Penambahan Data Kurikulum .....	129
5.2.5.2 Implementasi Pengubahan Data Kurikulum .....	130
5.2.5.3 Implementasi Penghapusan Data Kurikulum .....	131
5.2.6 Implementasi Perangkat Lunak untuk Administrasi Data <i>Login</i> .....	132
5.2.6.1 Implementasi Penambahan Data <i>Login</i> .....	132
5.2.6.2 Implementasi Pengubahan Data <i>Login</i> .....	133
5.2.6.3 Implementasi Penghapusan Data <i>Login</i> .....	134
5.2.6.4 Implementasi Pencarian Data <i>Login</i> .....	135
5.2.7 Implementasi Perangkat Lunak untuk Administrasi Data Peserta Mata Kuliah .....	135
5.2.7.1 Implementasi Penambahan Data Peserta Mata Kuliah .....	136
5.2.7.2 Implementasi Penghapusan Data Peserta Mata Kuliah .....	137
5.2.8 Implementasi Perangkat Lunak untuk Administrasi Data Jadwal Mata Kuliah .....	138
5.2.8.1 Implementasi Penambahan Data Jadwal Mata Kuliah .....	138
5.2.8.2 Implementasi Pengubahan Data Jadwal Mata Kuliah .....	139
5.2.8.3 Implementasi Penghapusan Data Jadwal Mata Kuliah .....	140
5.2.8.4 Implementasi Pencarian Data Jadwal Mata Kuliah .....	141
5.2.9 Implementasi Perangkat Lunak untuk Administrasi Data Jadwal Mata Kuliah Tambahan .....	143
5.2.9.1 Implementasi Penambahan Data Jadwal Mata Kuliah Tambahan .....	143
5.2.9.2 Implementasi Pengubahan Data Jadwal Mata Kuliah Tambahan .....	144
5.2.9.3 Implementasi Penghapusan Data Jadwal Mata Kuliah Tambahan .....	145
5.2.9.4 Implementasi Pencarian Data Jadwal Mata Kuliah Tambahan .....	146

5.2.10 Implementasi Perangkat Lunak untuk Administrasi Data Kehadiran Mahasiswa	147
5.2.10.1 Implementasi Penambahan Data Kehadiran Mahasiswa	147
5.2.10.2 Implementasi Pengubahan Data Kehadiran Mahasiswa	148
5.2.10.3 Implementasi Penghapusan Data Kehadiran Mahasiswa	149
5.2.10.4 Implementasi Pencarian Data Kehadiran Mahasiswa	150
5.2.11 Implementasi Perangkat Lunak untuk Pencetakan Kartu Kehadiran Mahasiswa (KKM)	152
5.2.12 Implementasi Perangkat Lunak untuk Pencetakan Kartu Kehadiran Dosen (KKD)	153
5.2.13 Implementasi Perangkat Lunak untuk Pencetakan <i>Report</i> Kehadiran Mahasiswa	154
5.3 Implementasi Antarmuka Aplikasi <i>Client</i>	155
<b>BAB VI PENGUJIAN DAN ANALISIS PERANGKAT LUNAK</b>	<b>160</b>
6.1 Pengujian Perancangan Basis Data	160
6.2 Pengujian Implementasi Aplikasi Sistem Secara <i>Client-Server</i>	163
6.2.1 Pengujian Performansi Koneksi	165
6.2.1.1 Pengujian Koneksi Basis Data	165
6.2.1.2 Pengujian Waktu Akses <i>Query</i>	168
6.2.2 Pengujian Aplikasi <i>Server</i>	170
6.2.2.1 Pengujian Proses <i>Login Server</i>	171
6.2.2.2 Pengujian Proses Administrasi Data Mahasiswa	173
6.2.2.2.1 Pengujian Proses Penambahan Data Mahasiswa	173
6.2.2.2.2 Pengujian Proses Pengubahan Data Mahasiswa	175
6.2.2.2.3 Pengujian Proses Penghapusan Data Mahasiswa	177
6.2.2.2.4 Pengujian Proses Pencarian Data Mahasiswa	178
6.2.2.3 Pengujian Proses Administrasi Data Dosen	180
6.2.2.3.1 Pengujian Proses Penambahan Data Dosen	180
6.2.2.3.2 Pengujian Proses Pengubahan Data Dosen	181
6.2.2.3.3 Pengujian Proses Penghapusan Data Dosen	183
6.2.2.3.4 Pengujian Proses Pencarian Data Dosen	185
6.2.2.4 Pengujian Proses Administrasi Data Mata Kuliah	186
6.2.2.4.1 Pengujian Proses Penambahan Data Mata Kuliah	186
6.2.2.4.2 Pengujian Proses Pengubahan Data Mata Kuliah	188
6.2.2.4.3 Pengujian Proses Penghapusan Data Mata Kuliah	189
6.2.2.4.4 Pengujian Proses Pencarian Data Mata Kuliah	191
6.2.2.5 Pengujian Proses Administrasi Data Kurikulum	192

6.2.2.5.1	Pengujian Proses Penambahan Data Kurikulum .....	193
6.2.2.5.2	Pengujian Proses Pengubahan Data Kurikulum .....	195
6.2.2.5.3	Pengujian Proses Penghapusan Data Kurikulum .....	196
6.2.2.6	Pengujian Proses Administrasi Data <i>Login</i> .....	198
6.2.2.6.1	Pengujian Proses Penambahan Data <i>Login</i> .....	198
6.2.2.6.2	Pengujian Proses Pengubahan Data <i>Login</i> .....	199
6.2.2.6.3	Pengujian Proses Penghapusan Data <i>Login</i> .....	201
6.2.2.6.4	Pengujian Proses Pencarian Data <i>Login</i> .....	203
6.2.2.7	Pengujian Proses Administrasi Data Peserta Mata Kuliah .....	204
6.2.2.7.1	Pengujian Proses Penambahan Data Peserta Mata Kuliah .....	204
6.2.2.7.2	Pengujian Proses Penghapusan Data Peserta Mata Kuliah .....	206
6.2.2.8	Pengujian Proses Administrasi Data Jadwal Mata Kuliah .....	208
6.2.2.8.1	Pengujian Proses Penambahan Data Jadwal Mata Kuliah .....	208
6.2.2.8.2	Pengujian Proses Pengubahan Data Jadwal Mata Kuliah .....	210
6.2.2.8.3	Pengujian Proses Penghapusan Data Jadwal Mata Kuliah .....	211
6.2.2.8.4	Pengujian Proses Pencarian Data Jadwal Mata Kuliah .....	213
6.2.2.9	Pengujian Proses Administrasi Data Kehadiran Mahasiswa .....	214
6.2.2.9.1	Pengujian Proses Penambahan Data Kehadiran Mahasiswa .....	214
6.2.2.9.2	Pengujian Proses Pengubahan Data Kehadiran Mahasiswa .....	216
6.2.2.9.3	Pengujian Proses Penghapusan Data Kehadiran Mahasiswa .....	217
6.2.2.9.4	Pengujian Proses Pencarian Data Kehadiran Mahasiswa .....	219
6.2.2.10	Pengujian Proses Pencetakan Kartu Kehadiran Mahasiswa .....	220
6.2.2.11	Pengujian Proses Pencetakan Kartu Kehadiran Dosen .....	222
6.2.2.12	Pengujian Proses Pencetakan <i>Report</i> Kehadiran Mahasiswa .....	223
6.2.3	Pengujian Aplikasi <i>Client</i> .....	224
6.2.3.1	Pengujian Proses <i>Login Client</i> .....	225
6.2.3.2	Pengujian Pencatatan Kehadiran Mahasiswa .....	228
<b>BAB VII</b>	<b>PENUTUP.....</b>	<b>231</b>
7.1	Kesimpulan .....	231
7.2	Saran .....	232
<b>DAFTAR PUSTAKA.....</b>		<b>233</b>
<b>LAMPIRAN.....</b>		<b>234</b>

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 2.1 Bentuk Tabel Dengan Kriteria <i>Boyce-Code Normal Form</i> .....	13
Tabel 2.2 Perubahan Struktur CDM ke PDM .....	23
Tabel 2.3 Tabel Kegunaan <i>Barcode</i> .....	30
Tabel 2.4 Tabel Karakter Code 39 Beserta Nilai Karakternya .....	33
Tabel 4.1 Tabel <i>Data Type Definition</i> Dari Tabel m_mhs.....	50
Tabel 4.2 Tabel <i>Data Type Definition</i> Dari Tabel m_dosen .....	50
Tabel 4.3 Tabel <i>Data Type Definition</i> Dari Tabel m_mk.....	51
Tabel 4.4 Tabel <i>Data Type Definition</i> Dari Tabel m_krk.....	51
Tabel 4.5 Tabel <i>Data Type Definition</i> Dari Tabel m_login.....	51
Tabel 4.6 Tabel <i>Data Type Definition</i> Dari Tabel m_hari .....	51
Tabel 4.7 Tabel <i>Data Type Definition</i> Dari Tabel d_perserta_mk .....	52
Tabel 4.8 Tabel <i>Data Type Definition</i> Dari Tabel d_jadwal .....	52
Tabel 4.9 Tabel <i>Data Type Definition</i> Dari Tabel d_hadir .....	52
Tabel 4.10 Tabel <i>Data Type Definition</i> Basis Data Kehadiran.....	53
Tabel 5.1 Hasil Pencarian Data Mata Kuliah.....	125
Tabel 5.2 Hasil Pencarian Data Mata Kuliah Lebih Dari Satu Kelas .....	141
Tabel 5.3 Hasil Pencarian Data Mata Kehadiran Mahasiswa .....	145
Tabel 5.4 Hasil Pencarian Data Mata Kehadiran Mahasiswa Dengan Tanggal Yang Berbeda.....	146
Tabel 6.1 Tabel Rata-rata Pengujian Waktu Akses <i>Query</i> .....	164
Tabel 6.2 Tabel Rata-rata Pengujian Waktu Akses per- <i>record</i> .....	165

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2.1 Siklus Informasi .....	10
Gambar 2.2 Level Abstraksi Data .....	15
Gambar 2.3 Model Data Hirarki .....	16
Gambar 2.4 Model Data Jaringan .....	16
Gambar 2.5 Model Data Relasional .....	17
Gambar 2.6 Sistem Tunggal ( <i>Stand-Alone</i> ) .....	18
Gambar 2.7 Basis Data Terpusat .....	19
Gambar 2.8 <i>Sistem Client-Server</i> Sederhana ( <i>two-tier</i> ) .....	19
Gambar 2.9 <i>Sistem Client-Server</i> Kompleks ( <i>N-tier</i> ) .....	20
Gambar 2.10 Konsep Kerja Sybase Power Designer 10 .....	22
Gambar 2.11 Arsitektur <i>Client-Server</i> Pada Borland Interbase .....	25
Gambar 2.12 Konsep <i>Client-Server</i> .....	26
Gambar 2.13 Tampilan IDE Delphi .....	28
Gambar 2.14 <i>Barcode</i> Tipe EAN .....	30
Gambar 2.15 <i>Barcode</i> Tipe Code 128 .....	31
Gambar 2.16 <i>Barcode</i> Tipe Code 93 .....	31
Gambar 2.17 <i>Barcode</i> Tipe Code 39 .....	32
Gambar 2.18 <i>Barcode</i> Tipe UPC .....	32
Gambar 2.19 <i>Layer</i> TPC/IP .....	35
Gambar 2.20 Unit-Unit Data Protocol Pada Arsitektur TCP/IP .....	36
Gambar 2.21 Topologi Bus .....	38
Gambar 2.22 Topologi Ring .....	38
Gambar 2.23 Topologi Star .....	39
Gambar 4.1 Diagram Pohon Perancangan Perangkat Lunak .....	42
Gambar 4.2 Diagram Pohon Analisis Kebutuhan Perangkat Lunak .....	43
Gambar 4.3 Analisis Administrasi Kehadiran Mahasiswa Secara Manual .....	46
Gambar 4.4 Analisis Sistem Informasi Kehadiran Mahasiswa .....	47
Gambar 4.5 Diagram Pohon Perancangan Sistem .....	48
Gambar 4.6 Diagram Blok Sistem .....	49
Gambar 4.7 Diagram Konteks .....	50
Gambar 4.8 Diagram Pohon Perancangan Basis Data .....	51
Gambar 4.9 <i>Entity Relationship Diagram</i> .....	51
Gambar 4.10 Desain Basis Data Kehadiran Ternormalisasi .....	56

Gambar 4.11 Diagram Pohon Perancangan Proses.....	57
Gambar 4.12 <i>Data Flow Diagram Level 0 Aplikasi Server</i> .....	58
Gambar 4.13 <i>Data Flow Diagram Level 0 Aplikasi Client</i> .....	59
Gambar 4.14 <i>Data Flow Diagram Level 1 Otentifikasi Login</i> .....	60
Gambar 4.15 <i>Data Flow Diagram Level 1 Administrasi Data Master</i> .....	61
Gambar 4.16 <i>Data Flow Diagram Level 1 Pencetakan Report dan Kartu Kehadiran</i> .....	63
Gambar 4.17 DFD Level 2 Proses 2.1 Administrasi Data Mahasiswa.....	64
Gambar 4.18 DFD Level 2 Proses 2.2 Administrasi Data Dosen.....	65
Gambar 4.19 DFD Level 2 Proses 2.3 Administrasi Data Mata Kuliah.....	66
Gambar 4.20 DFD Level 2 Proses 2.4 Administrasi Data Kurikulum.....	67
Gambar 4.21 DFD Level 2 Proses 2.5 Administrasi Data <i>Login</i> .....	68
Gambar 4.22 DFD Level 2 Proses 2.6 Administrasi Data Peserta Mata Kuliah.....	69
Gambar 4.23 DFD Level 2 Proses 2.7 Administrasi Data Jadwal Mata Kuliah.....	70
Gambar 4.24 DFD Level 2 Proses 2.8 Administrasi Data Kehadiran Mahasiswa.....	71
Gambar 4.25 <i>State Transition Diagram</i> Proses Pencatatan Kehadiran Mahasiswa.....	72
Gambar 4.26 Diagram Alir Aplikasi <i>Server</i> .....	73
Gambar 4.27 Diagram Alir Login <i>Server</i> .....	74
Gambar 4.28 Diagram Alir Administrasi Data Mahasiswa.....	75
Gambar 4.29 Diagram Alir Pencarian Data Mahasiswa.....	76
Gambar 4.30 Diagram Alir Administrasi Data Dosen.....	77
Gambar 4.31 Diagram Alir Pencarian Data Dosen.....	78
Gambar 4.32 Diagram Alir Administrasi Data Mata Kuliah.....	79
Gambar 4.33 Diagram Alir Pencarian Data Mata Kuliah.....	80
Gambar 4.34 Diagram Alir Administrasi Data Kurikulum.....	81
Gambar 4.35 Diagram Alir Administrasi Data <i>Login</i> .....	82
Gambar 4.36 Diagram Alir Pencarian Data <i>Login</i> .....	83
Gambar 4.37 Diagram Alir Administrasi Data Peserta Mata Kuliah.....	84
Gambar 4.38 Diagram Alir Administrasi Data Jadwal Mata Kuliah.....	85
Gambar 4.39 Diagram Alir Pencarian Data Jadwal Mata Kuliah.....	86
Gambar 4.40 Diagram Alir Administrasi Data Kehadiran Mahasiswa.....	87
Gambar 4.41 Diagram Alir Pencarian Data Kehadiran Mahasiswa.....	88
Gambar 4.42 Diagram Alir Pencetakan Kartu Kehadiran Mahasiswa (KKM).....	89
Gambar 4.43 Diagram Alir Pencetakan Kartu Kehadiran Dosen (KKD).....	90
Gambar 4.44 Diagram Alir Pencetakan <i>Report</i> Kehadiran Mahasiswa.....	91
Gambar 4.45 Diagram Alir Aplikasi <i>Client</i> .....	92
Gambar 4.46 Diagram Alir <i>Login Client</i> .....	93
Gambar 4.47 Diagram Alir Pencatatan Kehadiran Mahasiswa.....	94

Gambar 4.48 Diagram Pohon Perancangan Antarmuka .....	95
Gambar 4.49 Antarmuka Aplikasi <i>Server</i> Sebelum <i>Login</i> .....	95
Gambar 4.50 Antarmuka <i>Login Server</i> .....	96
Gambar 4.51 Antarmuka Administrasi Data Mahasiswa.....	97
Gambar 4.52 Antarmuka Administrasi Data Dosen .....	98
Gambar 4.53 Antarmuka Administrasi Data Mata Kuliah.....	99
Gambar 4.54 Antarmuka Administrasi Data Kurikulum .....	100
Gambar 4.55 Antarmuka Administrasi Data <i>Login</i> .....	101
Gambar 4.56 Antarmuka Administrasi Data Peserta Mata Kuliah .....	102
Gambar 4.57 Antarmuka Administrasi Data Jadwal Mata Kuliah.....	103
Gambar 4.58 Antarmuka Administrasi Data Jadwal Mata Kuliah Tambahan.....	104
Gambar 4.59 Antarmuka Administrasi Data Kehadiran Mahasiswa .....	105
Gambar 4.60 Antarmuka Pencetakan Kartu Kehadiran Mahasiswa .....	106
Gambar 4.61 Antarmuka Pencetakan Kartu Kehadiran Dosen.....	107
Gambar 4.62 Antarmuka Pencetakan <i>Report</i> Kehadiran Mahasiswa .....	108
Gambar 4.63 Antarmuka Aplikasi <i>Client</i> .....	109
Gambar 5.1 Diagram Pohon Implementasi Perangkat Lunak.....	110
Gambar 5.2 Implementasi Sistem Informasi Kehadiran Mahasiswa .....	110
Gambar 5.3 Implementasi Basis Data Kehadiran pada Borland Interbase .....	114
Gambar 5.4 Implementasi Antarmuka <i>Login Server</i> .....	115
Gambar 5.5 Implementasi Pesan Peringatan <i>Login Server</i> .....	115
Gambar 5.6 Implementasi Menu Tidak Aktif Sebelum Proses <i>Login</i> .....	116
Gambar 5.7 Implementasi Menu Aktif Setelah Proses <i>Login</i> .....	116
Gambar 5.8 Implementasi Penambahan Data Mahasiswa .....	117
Gambar 5.9 Implementasi Pengubahan Data Mahasiswa .....	118
Gambar 5.10 Implementasi Penghapusan Data Mahasiswa.....	119
Gambar 5.11 Implementasi Pencarian Data Mahasiswa.....	120
Gambar 5.12 Implementasi Penambahan Data Dosen.....	121
Gambar 5.13 Implementasi Pengubahan Data Dosen.....	122
Gambar 5.14 Implementasi Penghapusan Data Dosen .....	123
Gambar 5.15 Implementasi Pencarian Data Dosen.....	124
Gambar 5.16 Implementasi Penambahan Data Mata Kuliah .....	125
Gambar 5.17 Implementasi Pengubahan Data Mata Kuliah .....	126
Gambar 5.18 Implementasi Penghapusan Data Mata Kuliah .....	127
Gambar 5.19 Implementasi Pencarian Data Mata Kuliah.....	128
Gambar 5.20 Implementasi Penambahan Data Kurikulum.....	129
Gambar 5.21 Implementasi Pengubahan Data Kurikulum.....	130

Gambar 5.22 Implementasi Penghapusan Data Kurikulum.....	131
Gambar 5.23 Implementasi Penambahan Data <i>Login</i> .....	132
Gambar 5.24 Implementasi Pengubahan Data <i>Login</i> .....	133
Gambar 5.25 Implementasi Penghapusan Data <i>Login</i> .....	134
Gambar 5.26 Implementasi Pencarian Data <i>Login</i> .....	135
Gambar 5.27 Implementasi Penambahan Data Peserta Mata Kuliah.....	136
Gambar 5.28 Implementasi Penghapusan Data Peserta Mata Kuliah.....	137
Gambar 5.29 Implementasi Penambahan Data Jadwal Mata Kuliah.....	138
Gambar 5.30 Implementasi Pengubahan Data Jadwal Mata Kuliah.....	139
Gambar 5.31 Implementasi Penghapusan Data Jadwal Mata Kuliah.....	140
Gambar 5.32 Implementasi Pencarian Data Jadwal Mata Kuliah.....	141
Gambar 5.33 Implementasi Pencarian Data Jadwal Mata Kuliah lebih dari Satu Kelas.....	142
Gambar 5.34 Implementasi Penambahan Data Jadwal Mata Kuliah Tambahan.....	143
Gambar 5.35 Implementasi Pengubahan Data Jadwal Mata Kuliah Tambahan.....	144
Gambar 5.36 Implementasi Penghapusan Data Jadwal Mata Kuliah Tambahan.....	145
Gambar 5.37 Implementasi Pencarian Data Jadwal Mata Kuliah Tambahan.....	146
Gambar 5.38 Implementasi Penambahan Data Kehadiran Mahasiswa.....	147
Gambar 5.39 Implementasi Pengubahan Data Kehadiran Mahasiswa.....	148
Gambar 5.40 Implementasi Penghapusan Data Kehadiran Mahasiswa.....	149
Gambar 5.41 Implementasi Pencarian Data Kehadiran Mahasiswa.....	150
Gambar 5.42 Implementasi Pencarian Data Kehadiran Mahasiswa lebih dari Satu Jadwal.....	151
Gambar 5.43 Implementasi Pencetakan Kartu Kehadiran Mahasiswa.....	152
Gambar 5.44 Implementasi Kartu Kehadiran Mahasiswa.....	152
Gambar 5.45 Implementasi Pencetakan Kartu Kehadiran Dosen.....	153
Gambar 5.46 Implementasi Kartu Kehadiran Dosen.....	153
Gambar 5.47 Implementasi Pencetakan <i>Report</i> Kehadiran Mahasiswa.....	154
Gambar 5.48 Implementasi <i>Report</i> Kehadiran Mahasiswa.....	155
Gambar 5.49 Implementasi Antarmuka Kehadiran Mahasiswa.....	155
Gambar 5.50 Implementasi Proses <i>Login</i> Dosen.....	156
Gambar 5.51 Implementasi Hasil Proses <i>Login</i> Dosen.....	156
Gambar 5.52 Implementasi Proses <i>Logout</i> Dosen Berhasil Dilakukan.....	157
Gambar 5.53 Implementasi Antarmuka Aplikasi <i>Client</i> untuk Proses Pencatatan Kehadiran.....	157
Gambar 5.54 Halaman Antarmuka Hasil Pencatatan Kehadiran Mahasiswa.....	158
Gambar 5.55 Implementasi Pesan Peringatan <i>Login</i> .....	158
Gambar 5.56 Implementasi Pesan Peringatan Jadwal Mengajar.....	159
Gambar 5.57 Implementasi Pesan Peringatan Mata Kuliah.....	159
Gambar 5.58 Implementasi Pesan Peringatan Pencatatan Kehadiran.....	159

Gambar 6.1 Diagram Pohon Pengujian dan Analisis Perangkat Lunak.....	160
Gambar 6.2 <i>Conceptual Data Model Object</i> untuk Basis Data Kehadiran.....	161
Gambar 6.3 <i>Physical Data Model Object</i> untuk Basis Data Kehadiran .....	162
Gambar 6.4 <i>Script Generate Database</i> dari Basis Data Kehadiran .....	162
Gambar 6.5 Basis Data Kehadiran Hasil Proses <i>Generate</i> pada Sybase Power Designer 10 .....	163
Gambar 6.6 Diagram Blok Pengujian Secara <i>Client-Server</i> .....	164
Gambar 6.7 Diagram Pohon Pengujian Implementasi Aplikasi Secara <i>Client-Server</i> .....	164
Gambar 6.8 Proses yang Berjalan pada <i>Server</i> Sebelum ada Koneksi .....	166
Gambar 6.9 Proses yang Berjalan pada <i>Client</i> Sebelum ada Koneksi .....	167
Gambar 6.10 Proses yang Berjalan pada <i>Client</i> Setelah ada Koneksi .....	167
Gambar 6.11 Proses Pengujian Waktu Akses <i>Query</i> terhadap Basis Data Kehadiran.....	168
Gambar 6.12 Grafik Perbandingan Waktu Akses <i>Query</i> pada Basis Data Kehadiran.....	169
Gambar 6.13 Grafik Perbandingan Waktu Akses <i>Query</i> per-record pada Basis Data Kehadiran.	170
Gambar 6.14 Diagram Pohon Pengujian Aplikasi <i>Server</i> .....	171
Gambar 6.15 Halaman Antarmuka Proses <i>Login</i> .....	172
Gambar 6.16 Proses Login Administrator .....	172
Gambar 6.17 Menu Aktif Setelah Proses <i>Login</i> .....	172
Gambar 6.18 Menu Tidak Aktif Setelah Proses <i>Logout</i> .....	173
Gambar 6.19 Diagram Pohon Pengujian Proses Administrasi Data Mahasiswa .....	173
Gambar 6.20 Antarmuka Proses Penambahan Data Mahasiswa.....	174
Gambar 6.21 Hasil Penambahan Data Mahasiswa pada Basis Data Kehadiran .....	175
Gambar 6.22 Antarmuka Proses Pengubahan Data Mahasiswa .....	176
Gambar 6.23 Hasil Pengubahan Data Mahasiswa pada Basis Data Kehadiran .....	176
Gambar 6.24 Antarmuka Proses Penghapusan Data Mahasiswa.....	177
Gambar 6.25 Data Mahasiswa Sebelum Proses Penghapusan Data Mahasiswa .....	178
Gambar 6.26 Data Mahasiswa Setelah Proses Penghapusan Data Mahasiswa.....	178
Gambar 6.27 Antarmuka Proses Pencarian Data Mahasiswa .....	179
Gambar 6.28 Hasil Proses Pencarian Data Mahasiswa pada Basis Data Kehadiran .....	179
Gambar 6.29 Diagram Pohon Pengujian Proses Administrasi Data Dosen .....	180
Gambar 6.30 Antarmuka Proses Penambahan Data Dosen .....	181
Gambar 6.31 Hasil Penambahan Data Dosen pada Basis Data Kehadiran.....	181
Gambar 6.32 Antarmuka Proses Pengubahan Data Dosen .....	182
Gambar 6.33 Hasil Pengubahan Data Dosen pada Basis Data Kehadiran.....	183
Gambar 6.34 Antarmuka Proses Penghapusan Data Dosen.....	184
Gambar 6.35 Data Dosen Sebelum Proses Penghapusan Data Dosen.....	184
Gambar 6.36 Data Dosen Setelah Proses Penghapusan Data Dosen .....	184
Gambar 6.37 Antarmuka Proses Pencarian Data Dosen .....	185

Gambar 6.38 Hasil Proses Pencarian Data Dosen pada Basis Data Kehadiran .....	186
Gambar 6.39 Diagram Pohon Pengujian Proses Administrasi Data Mata Kuliah .....	186
Gambar 6.40 Antarmuka Proses Penambahan Data Mata Kuliah .....	187
Gambar 6.41 Hasil Penambahan Data Mata Kuliah pada Basis Data Kehadiran .....	188
Gambar 6.42 Antarmuka Proses Pengubahan Data Mata Kuliah .....	189
Gambar 6.43 Hasil Pengubahan Data Mata Kuliah pada Basis Data Kehadiran .....	189
Gambar 6.44 Antarmuka Proses Penghapusan Data Mata Kuliah .....	190
Gambar 6.45 Data Mata Kuliah Sebelum Proses Penghapusan Data Mata Kuliah .....	191
Gambar 6.46 Data Mata Kuliah Setelah Proses Penghapusan Data Mata Kuliah .....	191
Gambar 6.47 Antarmuka Proses Pencarian Data Mata Kuliah .....	192
Gambar 6.48 Hasil Proses Pencarian Data Mata Kuliah pada Basis Data Kehadiran .....	192
Gambar 6.49 Diagram Pohon Pengujian Proses Administrasi Data Kurikulum .....	193
Gambar 6.50 Antarmuka Proses Penambahan Data Kurikulum .....	194
Gambar 6.51 Hasil Penambahan Data Kurikulum pada Basis Data Kehadiran .....	194
Gambar 6.52 Antarmuka Proses Pengubahan Data Kurikulum .....	195
Gambar 6.53 Hasil Pengubahan Data Kurikulum pada Basis Data Kehadiran .....	196
Gambar 6.54 Antarmuka Proses Penghapusan Data Kurikulum .....	197
Gambar 6.55 Data Kurikulum Sebelum Proses Penghapusan Data Kurikulum .....	197
Gambar 6.56 Data Kurikulum Setelah Proses Penghapusan Data Kurikulum .....	197
Gambar 6.57 Diagram Pohon Pengujian Proses Administrasi Data <i>Login</i> .....	198
Gambar 6.58 Antarmuka Proses Penambahan Data <i>Login</i> .....	199
Gambar 6.59 Hasil Penambahan Data <i>Login</i> pada Basis Data Kehadiran .....	199
Gambar 6.60 Antarmuka Proses Pengubahan Data <i>Login</i> .....	200
Gambar 6.61 Hasil Pengubahan Data <i>Login</i> pada Basis Data Kehadiran .....	201
Gambar 6.62 Antarmuka Proses Penghapusan Data <i>Login</i> .....	202
Gambar 6.63 Data <i>Login</i> Sebelum Proses Penghapusan Data <i>Login</i> .....	202
Gambar 6.64 Data <i>Login</i> Setelah Proses Penghapusan Data <i>Login</i> .....	202
Gambar 6.65 Antarmuka Proses Pencarian Data <i>Login</i> .....	203
Gambar 6.66 Hasil Proses Pencarian Data <i>Login</i> pada Basis Data Kehadiran .....	204
Gambar 6.67 Diagram Pohon Pengujian Proses Administrasi Data Peserta Mata Kuliah .....	204
Gambar 6.68 Antarmuka Proses Penambahan Data Peserta Mata Kuliah .....	205
Gambar 6.69 Hasil Penambahan Data Peserta Mata Kuliah pada Basis Data Kehadiran .....	206
Gambar 6.70 Antarmuka Proses Penghapusan Data Peserta Mata Kuliah .....	207
Gambar 6.71 Data Peserta Mata Kuliah Sebelum Proses Penghapusan Data Peserta Mata Kuliah .....	207
Gambar 6.72 Data Peserta Mata Kuliah Setelah Proses Penghapusan Data Peserta Mata Kuliah .....	207
Gambar 6.73 Diagram Pohon Pengujian Proses Administrasi Data Jadwal Mata Kuliah .....	208
Gambar 6.74 Antarmuka Proses Penambahan Data Jadwal Mata Kuliah .....	209

Gambar 6.75 Hasil Penambahan Data Jadwal Mata Kuliah pada Basis Data Kehadiran ..	210
Gambar 6.76 Antarmuka Proses Pengubahan Data Jadwal Mata Kuliah .....	210
Gambar 6.77 Hasil Pengubahan Data Jadwal Mata Kuliah pada Basis Data Kehadiran...	211
Gambar 6.78 Antarmuka Proses Penghapusan Data Jadwal Mata Kuliah.....	212
Gambar 6.79 Data Jadwal Mata Kuliah Sebelum Proses Penghapusan Data Jadwal Mata Kuliah.....	212
Gambar 6.80 Data Jadwal Mata Kuliah Setelah Proses Penghapusan Data Jadwal Mata Kuliah.....	212
Gambar 6.81 Antarmuka Proses Pencarian Data Jadwal Mata Kuliah .....	213
Gambar 6.82 Hasil Proses Pencarian Data Jadwal Mata Kuliah pada Basis Data Kehadiran .....	214
Gambar 6.83 Diagram Pohon Pengujian Proses Administrasi Data Kehadiran Mahasiswa.....	214
Gambar 6.84 Antarmuka Proses Penambahan Data Kehadiran Mahasiswa .....	215
Gambar 6.85 Hasil Penambahan Data Kehadiran Mahasiswa pada Basis Data Kehadiran .....	216
Gambar 6.86 Antarmuka Proses Pengubahan Data Kehadiran Mahasiswa .....	217
Gambar 6.87 Hasil Pengubahan Data Kehadiran Mahasiswa pada Basis Data Kehadiran .....	217
Gambar 6.88 Antarmuka Proses Penghapusan Data Kehadiran Mahasiswa .....	218
Gambar 6.89 Data Kehadiran Mahasiswa Sebelum Proses Penghapusan Data Kehadiran Mahasiswa .....	219
Gambar 6.90 Data Kehadiran Mahasiswa Setelah Proses Penghapusan Data Kehadiran Mahasiswa .....	219
Gambar 6.91 Antarmuka Proses Pencarian Data Kehadiran Mahasiswa.....	220
Gambar 6.92 Hasil Proses Pencarian Data Kehadiran Mahasiswa pada Basis Data Kehadiran .....	220
Gambar 6.93 Antarmuka Proses Pencetakan Kartu Kehadiran Mahasiswa.....	221
Gambar 6.94 Kartu Kehadiran Mahasiswa .....	221
Gambar 6.95 Antarmuka Proses Pencetakan Kartu Kehadiran Dosen .....	222
Gambar 6.96 Kartu Kehadiran Dosen.....	223
Gambar 6.97 Antarmuka Proses Pencetakan <i>Report</i> Kehadiran Mahasiswa.....	224
Gambar 6.98 <i>Report</i> Kehadiran Mahasiswa .....	224
Gambar 6.99 Halaman Antarmuka Aplikasi <i>Client</i> untuk Proses <i>Login</i> .....	225
Gambar 6.100 Proses <i>Login</i> Dosen.....	226
Gambar 6.101 Pesan Peringatan Proses <i>Login</i> .....	226
Gambar 6.102 Proses Peringatan Jadwal Mengajar .....	227
Gambar 6.103 Proses <i>Login</i> Dosen Berhasil Dilakukan .....	227
Gambar 6.104 Proses <i>Logout</i> Dosen Berhasil Dilakukan .....	228
Gambar 6.105 Halaman Antarmuka Aplikasi <i>Client</i> untuk Proses Pencatatan Kehadiran .....	229
Gambar 6.106 Pesan Peringatan Mata Kuliah .....	229
Gambar 6.107 Halaman Antarmuka Aplikasi <i>Client</i> untuk Proses Pencatatan Kehadiran .....	230
Gambar 6.108 Pesan Peringatan Pencatatan Kehadiran.....	230

## DAFTAR ISTILAH

- Barcode** Metode dalam menggantikan karakter memakai susunan garis-garis vertikal dengan ketebalan tertentu berdasarkan sistem bilangan biner, sehingga *barcode* dapat secara langsung diakses dengan sistem yang dikontrol oleh komputer [IND-03:1].
- CDM** *Conceptual Data Model*, model data yang secara konseptual merepresentasikan struktur logika dari suatu basis data. Model konseptual ini terdiri dari beberapa objek yang akan digunakan untuk membentuk suatu basis data [POW-04].
- Client-Server** Arsitektur aplikasi *client-server*, yaitu basis data tidak tersebar di setiap komputer *client*. *Server* berada terpusat dan setiap *client* dapat mengakses *server* setiap saat menggunakan *interface* tertentu melalui jaringan yang sudah ada [DWI-03:1-2].
- Client Side** Aplikasi dijalankan pada sisi *client* dan pengolahan data dilakukan di *server* [DHA-02:70].
- Database** Sekumpulan data yang disusun dalam bentuk (beberapa) tabel yang saling berkaitan maupun berdiri sendiri. Data yang disimpan di dalam *database* (basis data) dapat diambil dan dimanipulasi dengan menggunakan *query* SQL [FAT-02:02].
- Delphi** Kompiler atau penerjemah bahasa Delphi yang awalnya dari bahasa Pascal yang merupakan bahasa tingkat tinggi sekelas dengan Basic [MAR-03].
- DFD** *Data Flow Diagram*, suatu model logika data atau proses yang dibuat untuk menggambarkan asal dan tujuan data dari suatu sistem, dimana data disimpan dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut [POH-97:16].
- ERD** *Entity Relationship Diagram*, menerjemahkan semesta data yang ada pada dunia nyata dengan memanfaatkan sejumlah perangkat konseptual [POH-97:35].
- Layer** Grup layanan, fungsi dan protokol yang lengkap dari sudut pandang konseptual, yakni di luar susunan hirarki yang menyusun grup serta yang terentang di seluruh sistem dan sesuai arsitektur jaringan [PUR-01:23].

- PDM** *Physical Data Model*, merupakan bentuk implementasi fisik dari suatu basis data. PDM merepresentasikan basis data lebih nyata daripada CDM [POW-04].
- Port** *Endpoint* pada sebuah koneksi *logical* pada jaringan TCP/IP. Nomor *port* menentukan jenis *port* [PUR-01:45].
- Query** (*Database*) Perintah atau pernyataan yang digunakan untuk mendefinisikan dan memanipulasi data di dalam *database* [FAT-02].
- Server Side** Aplikasi dijalankan pada sisi *server* dan pengolahan data juga dilakukan di *server* [DHA-02:70].
- SQL** *Structured Query Language*, bahasa pemrograman yang digunakan untuk memanipulasi data yang ada di dalam *database* [FAT-02].
- TCP/IP** *Transmission Control Protocol / Internet Protocol*, sekelompok protokol yang mengatur komunikasi data komputer di Internet. Protokol-protokol pada TCP/IP masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data [PUR-01:20].



## ABSTRAK

**SAID ISKANDAR. 2007. : Sistem Informasi Kehadiran Mahasiswa Dengan Menggunakan Barcode. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Ir. Primantara HT. dan Arief Andy Soebroto, ST., M.Kom.**

Pencatatan kehadiran mahasiswa yang konvensional memerlukan banyak intervensi pegawai bagian administrasi maupun kejujuran mahasiswa yang sedang dicatat kehadirannya. Hal ini sering memberikan peluang adanya manipulasi data kehadiran apabila pengawasan yang kontinyu pada proses ini tidak dilakukan semestinya. Sistem Informasi Kehadiran mahasiswa menggunakan *barcode* memberikan manfaat berupa proses pengambilan informasi kehadiran mahasiswa menjadi lebih mudah dan cepat. Administrasi presensi mahasiswa menjadi lebih mudah dengan bantuan aplikasi.

Perancangan dan pengimplementasian Sistem Informasi Kehadiran Mahasiswa dilakukan dengan menggunakan bahasa pemrograman Delphi (versi 7) dan basis data Borland Interbase (versi 6.5). Sistem Informasi Kehadiran Mahasiswa terdiri dari dua macam aplikasi sistem antara lain aplikasi *server* dan aplikasi *client*.

Pengujian Sistem Informasi Kehadiran Mahasiswa dilakukan pada setiap aplikasi sistem untuk mengetahui proses yang dilakukan oleh tiap-tiap aplikasi sistem tersebut. Hasil dari pengujian aplikasi sistem dapat diketahui bahwa aplikasi sistem pada Sistem Informasi Kehadiran Mahasiswa dapat melakukan proses sesuai dengan kegunaannya masing-masing. Pengujian juga dilakukan terhadap koneksi basis data dan waktu akses *query* yang digunakan oleh Sistem Informasi Kehadiran Mahasiswa. Sistem Informasi Kehadiran Mahasiswa yang dihubungkan dengan koneksi basis data Borland Interbase pada *port* 3050 dapat melakukan proses manipulasi terhadap data-data di dalam basis data.

Pengujian waktu akses *query* dilakukan pada masing-masing tabel dengan 2000, 4000, 6000, 8000, 10000 data *entry* dan 5 kali percobaan. Waktu akses *query* pada 2000 data *entry* adalah 0,28 s. Waktu akses *query* pada 4000 data *entry* adalah 0,55 s. Waktu akses *query* pada tabel 6000 data *entry* adalah 0,79 s. Waktu akses *query* pada 8000 data *entry* adalah 1,04 s. Waktu akses *query* 10000 data *entry* adalah 1,31 s. Rata-rata waktu akses *query* per-*record* adalah 0,000134 s. Sistem basis data Borland Interbase dapat berjalan stabil ditunjukkan dengan linieritas waktu akses *query* per-*record* terhadap masukan jumlah data yang linier. Hasil pengujian yang dilakukan terhadap koneksi basis data dan waktu akses *query* menunjukkan bahwa Sistem Informasi Kehadiran Mahasiswa dapat berfungsi dengan baik.

Kata Kunci : sistem informasi, pencatatan kehadiran, delphi, borland interbase, basis data, *barcode*, *server*, *client*.

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Evaluasi keberhasilan proses pendidikan meliputi dua hal yaitu proses pendidikan dan keberhasilan studi. Evaluasi keberhasilan proses pendidikan yang meliputi cara penyelenggaraan pendidikan, kesesuaian sarana dengan tujuan serta keikutsertaan mahasiswa dalam acara pendidikan. Evaluasi keberhasilan studi adalah evaluasi yang ditujukan untuk menentukan pengaruh ketidakteraturan penyelenggaraan acara pendidikan. Ketidakteraturan tersebut adalah mahasiswa atau dosen yang absen menentukan batalnya sebagian rencana dari tujuan penguasaan materi perkuliahan [NIM-00:17-18].

Kehadiran merupakan salah satu faktor yang menentukan prestasi akademik mahasiswa dalam kegiatan perkuliahan di perguruan tinggi. Pemahaman mahasiswa tentang tujuan suatu mata kuliah akan tercapai bila dosen dan mahasiswa dapat menciptakan suasana interaktif dalam kelas. Peran mahasiswa seharusnya lebih dominan dalam membangun dialog karena itu mahasiswa dituntut untuk belajar secara mandiri. Dosen berperan dalam mengarahkan mahasiswa sehingga mahasiswa dapat mengerti sesuai dengan tujuan mata kuliah yang diajarkan. Berbeda dengan sekolah di mana guru lebih dominan dalam mencapai tujuan dari proses belajar-mengajar.

Kewajiban untuk mengikuti minimal delapan puluh persen dari jumlah tatap muka dengan dosen pengajar dalam satu mata kuliah sebagai syarat mengikuti ujian akhir semester merupakan salah satu kebijakan di Jurusan Elektro Universitas Brawijaya. Kebijakan tersebut membantu mahasiswa dalam mencapai tujuan mata kuliah yang ditempuh. Kehadiran mahasiswa dalam kegiatan perkuliahan menjadi hal mutlak yang diperlukan agar tercipta pemahaman dari materi mata kuliah yang disampaikan oleh dosen pengajar. Pengisian kehadiran menggunakan tanda-tangan merupakan salah satu cara menjamin kehadiran mahasiswa untuk mengikuti mata kuliah yang telah direncanakan dalam KRS (Kartu Rencana Studi). Mahasiswa menyiasati hal tersebut dengan melakukan titip absen. Titip absen merupakan suatu bentuk kegiatan dimana seorang mahasiswa yang tidak masuk kuliah menitipkan pesan kepada mahasiswa lainnya untuk mengisikan kehadiran dirinya pada suatu kegiatan perkuliahan.

Presensi mahasiswa yang konvensional memerlukan banyak intervensi pegawai bagian administrasi maupun kejujuran mahasiswa yang sedang dicatat kehadirannya. Hal ini sering memberikan peluang adanya manipulasi data kehadiran apabila pengawasan yang kontinyu pada proses ini tidak dilakukan semestinya.

Proses presensi mahasiswa merupakan proses yang repetitif. Mahasiswa mengikuti kuliah pada waktu tertentu dan membubuhkan tanda-tanggannya setiap kali mengikuti perkuliahan. Dosen akan memeriksa jumlah kehadiran masing-masing mahasiswa yang mengikuti mata kuliah yang diajarkannya untuk kelayakan dalam mengikuti ujian akhir semester. Prosedur tersebut diulang-ulang terus menerus tanpa banyak perubahan. Pengulangan prosedur tersebut dapat lebih dipermudah dengan program dan basis data yang akan menangani permasalahan di atas [CAH-02].

Aplikasi yang akan dibuat ini adalah aplikasi perangkat lunak berbasis aplikasi basis data yang memberi kecepatan dan kemudahan dalam memasukkan data presensi mahasiswa. Identitas mahasiswa berupa nim tersimpan dalam bentuk *barcode*. Mahasiswa melakukan proses presensi dengan menunjukkan kartu kehadiran mahasiswa pada *scanner barcode*.

Aplikasi Sistem Informasi Kehadiran Mahasiswa (SIKMA) menggunakan Borland Delphi untuk membangun *form* aplikasi dan Borland Interbase sebagai *server* basis data. Aplikasi akan dijalankan pada LAN (*Local Area Network*) menggunakan protokol TCP/IP.

Borland Delphi menggunakan bahasa Pascal yang bersifat visual sehingga memudahkan pengguna dalam membangun *form* aplikasi. Borland Delphi termasuk keluarga pemrograman visual seperti Visual Basic dan Visual C. Pemrograman visual dapat diartikan perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram dapat memilih objek yang ingin dimasukkan ke dalam *form*, kemudian tingkah laku objek saat menerima aksi (*event*) dapat dibuat programnya [TED-03].

Borland Interbase memiliki keutamaan dalam optimasi *query* secara otomatis oleh *server*, simultan akses ke banyak basis data, dan menyediakan *tool* yang interaktif untuk pendefinisian data dan *query* [DWI-03:3].

Sistem Informasi Kehadiran Mahasiswa menggunakan *barcode* memberikan manfaat berupa proses pengambilan informasi kehadiran mahasiswa menjadi lebih mudah dan cepat. Administrasi presensi mahasiswa menjadi lebih mudah dengan bantuan aplikasi. Tujuan lain dari membuat SIKMA adalah membangun kedisiplinan mahasiswa dan dosen sehingga tujuan perkuliahan akan tercapai.

## 1.2 Rumusan Masalah

Berdasar latar belakang di atas, maka dapat disusun rumusan masalah sebagai berikut.

1. Merancang dan membuat Sistem Informasi Kehadiran Mahasiswa.
2. Merancang dan membuat antarmuka Sistem Informasi Kehadiran Mahasiswa menggunakan *software* pemrograman Borland Delphi 7 dan *software* basis data Borland Interbase 6.5.
3. Merancang dan membuat Kartu Kehadiran Mahasiswa yang berisikan nomor induk mahasiswa dalam bentuk *barcode*.

## 1.3 Batasan Masalah

Pada skripsi ini diperlukan pembatasan masalah antara lain adalah:

1. Sistem Informasi Kehadiran Mahasiswa terdiri dari aplikasi *server* dan aplikasi *client*.
2. Perancangan kartu kehadiran mahasiswa menggunakan *barcode* dengan tipe Code-39.
3. *Scanner barcode* yang digunakan adalah Zebex tipe Z-3080.
4. Aplikasi dibuat dengan menggunakan arsitektur *client-server* dan diterapkan pada jaringan intranet dengan protokol TCP/IP.
5. Model kehadiran yang digunakan sebagai acuan adalah kehadiran mahasiswa Jurusan Elektro Universitas Brawijaya.

## 1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah merancang dan membuat aplikasi Sistem Informasi Kehadiran Mahasiswa menggunakan *barcode*.

Sistem Informasi Kehadiran Mahasiswa diharapkan memberikan manfaat bagi penulis dan Jurusan Elektro Universitas Brawijaya.

Manfaat untuk penulis antara lain:

1. Dapat menerapkan ilmu yang diperoleh dari sistem perkuliahan pada Teknik Elektro Konsentrasi Sistem Informatika dan Komputer Universitas Brawijaya.
2. Dapat merancang Sistem Informasi Kehadiran Mahasiswa yang sesuai dengan tujuannya.

Manfaat untuk Jurusan Elektro Universitas Brawijaya:

1. Administrasi mahasiswa, dosen, jadwal, peserta mata kuliah yang terpusat.
2. Mempermudah proses presensi mahasiswa dengan menggunakan *barcode*.
3. Meringankan pekerjaan administrasi dalam mengatur acara perkuliahan.

## 1.5 Sistematika Penulisan

### BAB I Pendahuluan

Membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

### BAB II Teori Dasar

Membahas teori dasar dan penunjang mengenai sistem informasi, basis data, basis data Borland Interbase, pemrograman Borland Delphi, konsep *barcode*, dan jaringan komputer.

### BAB III Metodologi penulisan

Membahas tentang metode yang digunakan dalam pembahasan skripsi. Metode yang digunakan antara lain studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis perangkat lunak serta pengambilan kesimpulan dan saran.

### BAB IV Perancangan Perangkat Lunak

Membahas mengenai perancangan sistem, perancangan basis data, perancangan proses, dan perancangan antarmuka Sistem Informasi Kehadiran Mahasiswa.

### BAB V Implementasi Perangkat Lunak

Membahas tentang implementasi perancangan basis data dan implementasi antarmuka Sistem Informasi Kehadiran Mahasiswa.

### BAB VI Pengujian Dan Analisis

Membahas tentang pengujian basis data dan pengujian implementasi Sistem Informasi Kehadiran Mahasiswa.

### BAB VII Kesimpulan dan Saran

Membahas tentang kesimpulan dan saran-saran yang diperlukan untuk mengembangkan aplikasi selanjutnya.

## BAB II

### TEORI DASAR

Bab ini menjelaskan teori dasar yang digunakan untuk menunjang penulisan skripsi mengenai sistem informasi kehadiran mahasiswa dengan menggunakan *barcode*. Teori dasar yang digunakan, yaitu sistem informasi, basis data, pemrograman Borland Delphi, jaringan komputer, dan *barcode*.

#### 2.1 Sistem Informasi

Teori dasar sistem informasi meliputi definisi sistem informasi, siklus informasi, perangkat pemodelan sistem informasi.

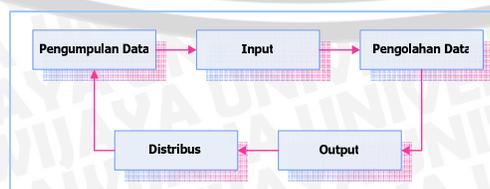
Sistem informasi didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, dan menyimpan serta mendistribusikan informasi. Sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi [DHA-02:11].

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna bagi pengguna sistem informasi. Informasi adalah data yang menggambarkan suatu kejadian. Definisi data secara rinci adalah sebagai berikut [DHA-02:7]:

- Data adalah penggambaran dari kejadian yang dihadapi.
- Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Dimana kesatuan nyata (*fact and entity*) merupakan suatu obyek nyata seperti tempat, barang, dan orang yang betul-betul ada dan terjadi.
- Data adalah sesuatu yang nyata atau setengah nyata yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian.

##### 2.1.1 Siklus Informasi

Pengolahan data menjadi informasi merupakan suatu siklus, yang terdiri dari tahap-tahap berikut [DHA-02:12]:



**Gambar 2.1** Siklus Informasi  
Sumber: [DHA-02:13]

- Pengumpulan data

Pengumpulan data adalah suatu proses pengambilan data asli dengan berbagai cara, seperti sampling data transaksi, menggunakan formulir, dan lain sebagainya yang biasanya merupakan proses pencatatan data ke dalam komputer.

- *Input*

*Input* merupakan proses untuk memasukan data yang telah dikumpulkan ke dalam komputer.

- Pengolahan data

Pada tahap ini data diolah sesuai dengan prosedur yang telah diberikan. Prosedur pengolahan data tersebut merupakan urutan langkah untuk mengolah data yang ditulis dalam suatu bahasa pemrograman. Kegiatan pengolahan data meliputi:

- Pencatatan data masukan (*recording/capturing*)

Data harus dicatat pada prosedur dalam suatu sistem sebelum data diproses menjadi informasi yang lebih berguna. Pencatatan data merupakan proses awal dari jalannya sistem informasi sehingga kelengkapan pencatatan data berpengaruh terhadap kualitas informasi yang dihasilkan.

- Manipulasi data

Manipulasi data merupakan suatu operasi yang dilakukan terhadap data untuk membentuk suatu informasi yang dapat dikelompokkan. Operasi manipulasi data dapat berupa pengelompokan, perhitungan, pengurutan, penggabungan, meringkas, menyimpan, dan menggunakannya kembali.

- *Output*

Hasil pengolahan data akan ditampilkan pada monitor atau dicetak pada printer sebagai informasi yang berguna.

- Distribusi

Data tersebut dapat didistribusikan kepada pihak yang berkepentingan atau pengguna akhir dalam sistem informasi setelah diolah menjadi informasi yang berguna. Pendistribusian data dapat berupa pengumuman mengenai informasi dari hasil keputusan.

### 2.1.2 Perangkat Pemodelan Sistem Informasi

Tiga alasan yang menyebabkan pemodelan sistem sebaiknya dilakukan, yaitu [POH-97:9]:

- Memfokuskan perhatian pada hal-hal penting dalam sistem tanpa mesti terlibat terlalu jauh.
- Mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai dengan resiko dan biaya minimal.
- Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendesain sistem dan pemrogram membangun sistem.

Pemodelan sistem dapat dipresentasikan dengan berbagai cara antara lain, *Data Flow Diagram Context Level*, *Data Flow Diagram Levelled*, *State-Transition Diagram*, dan banyak model lainnya.

#### 2.1.2.1 Data Flow Diagram Context Level (Diagram Konteks)

Diagram konteks adalah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem [POH-97:11].

Diagram konteks mempunyai karakteristik penting dari sebuah sistem, yaitu:

1. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi yang disebut juga sebagai terminator.
2. Data masuk, data yang diterima sistem dari lingkungan dan harus diproses dengan cara-cara tertentu.
3. Data keluar, data yang dihasilkan sistem dan diberikan ke pihak luar.
4. Penyimpanan data yang digunakan secara bersama-sama antara sistem dan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya, dibuat oleh lingkungan dan digunakan oleh sistem.
5. Batasan antara sistem dan lingkungan.

### 2.1.2.2 Data Flow Diagram Levelled (DFD Levelled)

DFD *Levelled* menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan proses penyimpanan serta aliran data. Ada empat komponen dalam model ini yaitu [POH-97:16-18]:

1. Proses, fungsi dan tranformasi. Proses ini menunjukkan tranformasi dari masukan menjadi keluaran, dalam hal ini bisa jadi hanya satu keluaran.
2. Aliran data digunakan untuk menggambarkan gerakan paket data informasi dari satu bagian ke bagian lain..
3. Penyimpanan digunakan untuk memodelkan kumpulan data atau paket data.
4. Terminator mewakili entitas luar dimana sistem berkomunikasi.

*Data Flow Diagram* digunakan untuk menggambarkan sistem secara logika tanpa mempertimbangkan logika fisik. *Data Flow Diagram* terdiri dari empat simbol, yaitu [JOG-99:699]:

1. Entitas eksternal

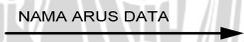
Setiap elemen mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima masukan dan menghasilkan keluaran kepada lingkungan luarnya.



NAMA ENTITAS

2. Arus data

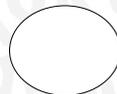
Arus data digambarkan dengan simbol panah, mengalir di antara proses, simpanan data dan entitas eksternal. Arus menunjukkan aliran dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.



NAMA ARUS DATA

3. Proses

Proses merupakan kegiatan yang dilakukan oleh organisasi, mesin, atau komputer dari hasil arus data yang masuk ke dalam proses untuk menghasilkan arus data yang akan keluar dari proses. Proses terjadi karena adanya arus data yang masuk dan hasil dari proses juga merupakan arus data yang mengalir ke suatu proses lain yang menerima lebih dari satu arus data dan juga menghasilkan sebuah arus data. Proses dilambangkan dengan simbol lingkaran.



Beberapa hal yang harus dihindari dalam menggambarkan suatu proses, yaitu:

- a. Proses yang mempunyai masukan tetapi tidak menghasilkan keluaran, kesalahan ini disebut *Black Hole*.
  - b. Proses yang menghasilkan keluaran tetapi tidak mendapatkan masukan, kesalahan ini disebut *Miracle*.
  - c. Proses yang menghasilkan keluaran lengkap tetapi mendapat masukan dengan data terbatas atau masukan kurang lengkap, kesalahan ini disebut *Grey Hole*.
4. Penyimpanan Data

Penyimpanan data digunakan untuk menyimpan data hasil proses maupun menyediakan data untuk diproses pada waktu lain. Penyimpanan data dilambangkan dengan simbol:



Beberapa hal yang perlu diperhatikan dalam menggambarkan penyimpanan data, antara lain:

- a. Proses merupakan satu-satunya yang berhubungan dengan penyimpanan data karena proses yang menggunakan data atau mengubah data di penyimpanan.
- b. Arus data yang menuju ke penyimpanan data dari suatu proses menunjukkan proses *update* terhadap data yang tersimpan di penyimpanan data.
- c. Arus data yang berasal dari penyimpanan data ke proses menunjukkan bahwa arus tersebut menggunakan data yang terdapat pada penyimpanan data.

### 2.1.2.3 State-Transition Diagram (STD)

*State-Transition Diagram* (STD) memodelkan tingkah laku ketergantungan sistem pada waktu. STD merupakan model yang bagus untuk mendeskripsikan kebutuhan pemodelan *real-time system* [POH-97:58].

Dua pendekatan yang diperlukan dalam membuat STD antara lain:

- Memulai mengidentifikasi keadaan yang mungkin, memisah-misahkannya dalam diagram.
- Memulai dengan kondisi awal (*initial state*), dan dengan sistematis melacak jalan menuju keadaan berikutnya.

## 2.2 Basis Data

Teori dasar basis data meliputi definisi basis data, perancangan basis data, Sistem Manajemen Basis Data, SQL (*Structured Query Language*), Sybase Power Designer 10, dan basis data Borland Interbase.

Basis data terdiri atas dua kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya. Data direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [FAT-02:2].

Basis data merupakan kumpulan dari item data yang saling berhubungan dan diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dimanipulasi dengan *software* untuk kegunaan tertentu [IRM-03].

Pengertian basis data tersebut di atas masih sangat umum. Penggunaan istilah basis data lebih dibatasi pada arti yang khusus, yaitu [ELM-94]:

- a. Basis data merupakan penyajian suatu aspek dari dunia nyata, misalnya basis data perbankan, perpustakaan, pertanahan, perpajakan dan sebagainya.
- b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
- c. Basis data perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

### 2.2.1 Perancangan Basis Data

Perancangan basis data diperlukan agar basis data yang diinginkan menjadi basis data yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam manipulasi data [FAT-02:39]. Metode yang digunakan dalam perancangan basis data, yaitu *Entity Relationship Diagram*, dan normalisasi.

#### 2.2.1.1 *Entity-Relationship Diagram*

Diagram *Entity-Relationship* menerjemahkan semesta data yang ada pada dunia nyata dengan memanfaatkan sejumlah perangkat konseptual [POH-97:35]. Komponen-komponen pembentuk diagram *Entity-Relationship* tersebut, yaitu:

a. Entitas dan Himpunan Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas. Entitas menunjuk pada individu suatu objek, sedangkan himpunan entitas menunjuk pada rumpun dari individu tersebut. Pada diagram *Entity-Relationship*, himpunan entitas digambarkan dengan persegi panjang [FAT-02:73].



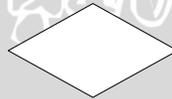
b. Atribut

Setiap entitas memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting dalam pembentukan model data. Pada diagram *Entity-Relationship*, atribut digambarkan dengan elips [FAT-02:74].



c. Relasi dan Himpunan Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas tersebut membentuk suatu himpunan relasi. Pada diagram *Entity-Relationship*, himpunan relasi digambarkan dengan belah ketupat [FAT-02:75].



d. Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misal himpunan A dan himpunan B) dapat berupa [FAT-02:77-79]:

- Satu ke Satu (*One to One*)  
Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, begitu pula sebaliknya.
- Satu ke Banyak (*One to Many*)  
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.
- Banyak ke Banyak (*Many to Many*)  
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, begitu pula sebaliknya.

Diagram *Entity-Relationship* selalu dibuat bertahap. Tujuan dari pentahapan ini adalah untuk mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau. Langkah-langkah yang harus dilakukan dalam membuat diagram *Entity-Relationship*, yaitu [FAT-02:84]:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta *foreign key*-nya.
4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non-key*).

### 2.2.1.2 Normalisasi

Normalisasi merupakan cara pendekatan lain dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data. Normalisasi menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal [FAT-02:39].

Sebuah tabel dapat dikategorikan baik dan efisien atau normal, jika telah memenuhi tiga kriteria berikut [FAT-02:50]:

- Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*).

- Terpeliharanya Ketergantungan Fungsional antar atribut pada saat perubahan data (*Dependency Preservation*).
- Tidak melanggar *Boycode Normal Form* (BCNF).

Jika kriteria ke tiga tidak dapat dipenuhi, maka setidaknya tabel yang akan dinormalisasi tidak melanggar bentuk normal tahap tiga (*3<sup>th</sup> Normal Form*).

#### 2.2.1.2.1 *Lossless-Join Decomposition*

Salah satu cara untuk membentuk tabel dengan struktur yang baik dan efisien adalah dengan mendekomposisi tabel tersebut. Dekomposisi yang benar adalah ketika sebuah tabel yang didekomposisi kemudian digabungkan kembali, maka tabel tersebut akan sama dengan keadaan sebelum didekomposisi [FAT-02:54].

#### 2.2.1.2.2 *Dependency Preservation*

Kriteria kedua untuk membentuk tabel yang efisien adalah dengan melakukan *Dependency Preservation* atau pemeliharaan ketergantungan. Perubahan yang terjadi terhadap data harus menjamin tidak akan mempengaruhi ketergantungan fungsional terhadap atribut-atribut yang terdapat dalam tabel tersebut [FAT-02:56].

#### 2.2.1.2.3 *Boyce-Code Normal Form (BCNF)*

Tabel dikatakan memenuhi kriteria *Boyce-Code Normal Form* jika semua ketergantungan fungsional dengan notasi  $x \rightarrow y$ , maka  $x$  harus menjadi *super key* pada tabel tersebut. Suatu tabel dengan atribut `nim`, `nama`, `kode_mk`, dan `nama_mk`. Tabel tersebut apabila didekomposisi akan didapatkan bentuk:

$$\text{nim} \rightarrow \text{nama}, \text{kode\_mk}, \text{nama\_mk}$$

akan tetapi bentuk tersebut belum memenuhi kriteria *Boyce-Code Normal Form*. Kriteria *Boyce-Code Normal Form* dapat terpenuhi dengan melakukan dekomposisi kembali menjadi dua tabel seperti dalam Tabel 2.1 yaitu [FAT-02:57]:

$$\text{nim} \rightarrow \text{nama}$$

$$\text{kode\_mk} \rightarrow \text{nama\_mk}$$

**Tabel 2.1** Bentuk tabel dengan kriteria *Boyce-Code Normal Form*

nim	Nama	kode_mk	Nama_mk
0001060282	Ardiansyah A.	TKE-101	Struktur Data
0001060373	Said Iskandar	TKE-102	Sistem Operasi
0001060400	Wawan R.	TKE-104	DTE

#### 2.2.1.2.4 Bentuk Normal Tahap Pertama (*1<sup>st</sup> Normal Form*)

Bentuk normal tahap pertama terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*Multivalued Attribute*) atau lebih dari satu atribut dengan domain nilai yang sama [FAT-02:65].

#### 2.2.1.2.5 Bentuk Normal Tahap Kedua (*2<sup>nd</sup> Normal Form*)

Bentuk normal tahap kedua terpenuhi jika pada sebuah tabel, semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah tabel dikatakan tidak memenuhi bentuk normal tahap kedua, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*) [FAT-02:67].

#### 2.2.1.2.6 Bentuk Normal Tahap Ketiga (*3<sup>rd</sup> Normal Form*)

Bentuk normal tahap ketiga sudah dapat membentuk sebuah tabel yang baik dan efisien. Bentuk normal tahap ketiga merupakan kriteria alternatif jika kriteria BCNF tidak dapat terpenuhi. Sebuah tabel dikatakan berada dalam bentuk normal tahap ketiga jika setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka [FAT-02:63]:

- X merupakan *super key* pada tabel tersebut atau
- A merupakan bagian dari *key primary* pada tabel tersebut.

### 2.2.2 Sistem Manajemen Basis Data

Sistem Manajemen Basis Data adalah kumpulan program yang digunakan untuk membuat dan mengelola basis data. Sistem Manajemen Basis Data merupakan sistem perangkat lunak yang secara umum dapat digunakan untuk melakukan pemrosesan dalam hal pendefinisian, penyusunan, dan manipulasi basis data untuk berbagai aplikasi. Pendefinisian basis data meliputi spesifikasi tipe data, struktur dan pembatasan (*constraint*) dari data yang harus disimpan dalam basis data. Penyusunan basis data meliputi proses memasukkan data dalam media penyimpanan data. Proses memasukkan data diatur oleh Sistem Manajemen Basis Data. Manipulasi basis data meliputi pembuatan pertanyaan (*query*) dari basis data untuk mendapatkan informasi tertentu, melakukan pembaharuan (*updating*) data, dan pembuatan laporan (*report generation*) dari data dalam basis data [WAL-02:2].

### 2.2.2.1 Abstraksi Data

Salah satu tujuan dari Sistem Manajemen Basis Data adalah untuk menyediakan fasilitas atau antarmuka dalam melihat data kepada pemakai [FAT-02:13-14].

Ada tiga level abstraksi data:

1. Level Fisik (*Physical Level*)

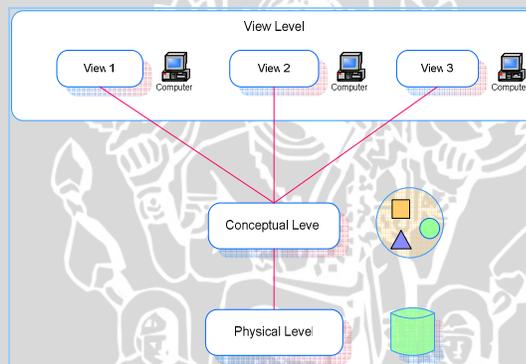
Level fisik merupakan level terendah dalam abstraksi data yang menunjukkan bagaimana sesungguhnya data disimpan.

2. Level Logik (*Conceptual Level*)

Level logik merupakan level berikutnya dalam abstraksi data yang menggambarkan data secara fungsional.

3. Level Penampakan (*View Level*)

Level penampakan merupakan level tertinggi dari abstraksi data yang menunjukkan sebagian dari basis data.



**Gambar 2.2** Level Abstraksi Data  
Sumber: [FAT-02:14]

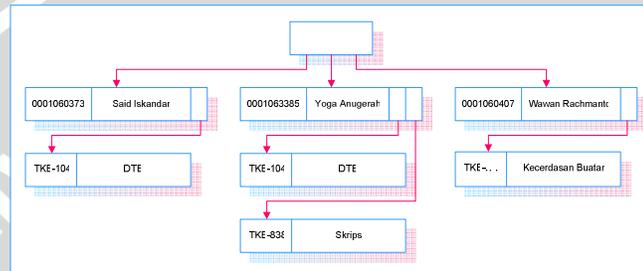
### 2.2.2.2 Model Data

Model data didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, makna data, dan batasan data [FAT-02:69]. Model data digunakan untuk merancang basis data yang menggambarkan keadaan nyata dalam suatu sistem. Pada sub-bab berikut ini dijelaskan secara singkat model data hirarki, jaringan, dan relasional.

#### 2.2.2.2.1 Model Data Hirarki

Model data hirarki menggambarkan kumpulan *record* yang dihubungkan satu sama lain melalui hubungan berdasarkan *pointer* yang membentuk struktur pohon.

Model data hirarki memiliki kelemahan, karena memungkinkan terjadinya redundansi (perulangan data) yang banyak pada *record* derajat berikutnya. Pada ilustrasi di bawah tergambar mahasiswa yang melakukan registrasi untuk mengambil mata kuliah dalam satu semester. *Record* mata kuliah yang sama harus ditulis ulang, ketika diambil oleh mahasiswa yang berbeda. Model data hirarki mempunyai fleksibilitas yang sangat rendah dan kompleks dalam menambah dan menyisipkan *record* baru. Pemrograman pun menjadi sangat kompleks, meskipun sebenarnya proses pengorganisasian data pada model ini efisien [DHA-02:107].

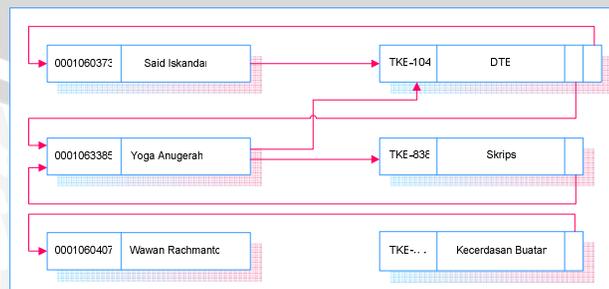


**Gambar 2.3** Model Data Hirarki  
Sumber: [DHA-02:108]

#### 2.2.2.2.2 Model Data Jaringan

Model data jaringan merupakan model data yang terdiri atas sekumpulan *record* yang dihubungkan melalui *pointer* yang membentuk relasi antar *record* dalam bentuk *ring*.

Model data jaringan memiliki kelemahan, yaitu tidak memungkinkan relasi banyak ke banyak (*many to many*). Model data jaringan mempunyai fleksibilitas yang sangat rendah dan kompleks dalam menambah atau menyisipkan *record*. Pemrogramannya menjadi sangat kompleks meskipun model ini menjanjikan efisiensi dalam proses pengorganisasian data dan menjamin tidak terjadinya redundansi [DHA-02:108].



**Gambar 2.4** Model Data Jaringan  
Sumber: [DHA-02:108]

### 2.2.2.2.3 Model Data Relasional

Model data relasional berisi kumpulan tabel, di mana setiap tabel mempunyai nama dan struktur yang unik. Masing-masing *record* data diorganisasikan dalam struktur yang sama pada setiap tabel. *Record* data memiliki field kunci yang akan menjadi penghubung antar tabel yang ada dan berkait satu sama lain [DHA-02:108].

Model data relasional tidak mempunyai hirarki atau jenjang dalam medan rekaman data, dan setiap medan data dapat dijadikan kunci data. Data rekaman disusun dari nilai yang berhubungan disebut baris. Baris-baris ini akan tersusun membentuk satu tabel, yang biasanya tersimpan dalam satu berkas. Tabel-tabel ini secara keseluruhan merupakan penyajian dari atribut data yang saling berhubungan.

Pencarian medan atribut dari suatu tabel atau banyak tabel dapat dilakukan dengan cepat dengan menggunakan model data relasional. Model data relasional tidak mempunyai tambahan dalam jenis pertanyaan (*query*) dengan menggunakan operasi gabungan yang panjang [WAL-02:28].

No	Tanggal	NIM	Kode MK
1	30-01-2006	0001060373	TKE-104
2	30-01-2006	0001060407	TKE-355
3	31-01-2006	0001063385	TKE-104

NIM	Nama	Alamat
0001060373	Said Iskandar	Tlogomas
0001060407	Wawan R	Joyogranc
0001063385	Yoga A	Ciwular

Kode MK	Nama MK	SKS
TKE-104	DTE	2
TKE-355	Kecerdasan Buatan	2
TKE-104	Sistem Operas	3

**Gambar 2.5** Model Data Relasional  
Sumber: [DHA-02:109]

Model data relasional memberikan kelebihan tersendiri dibandingkan dengan dua model sebelumnya, antara lain [DHA-02:109]:

1. Kemudahan dalam pembentukan struktur data masing-masing *file*.
2. Kompleksitas untuk mengaitkan antar tabel tidak terjadi karena hubungan antar tabel ditentukan oleh *field* kunci yang telah ditetapkan sebagai penghubung antar *file*.

3. Pemrograman menjadi sederhana, sedangkan tingkat fleksibilitas dalam mengorganisasikan data sangat tinggi.

Model relasional mempunyai kelemahan, antara lain adalah [WAL-02:30]:

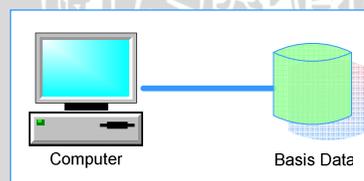
1. Penerapan model ini lebih sulit terutama untuk data dalam jumlah yang besar dan tingkat kerumitannya tinggi.
2. Proses pencarian informasi lebih lambat, disebabkan beberapa tabel tidak dihubungkan secara fisik seperti pada model hirarki dan jaringan. Manipulasi data yang menggunakan beberapa tabel akan memerlukan waktu yang lebih lama, karena tabel-tabel harus dihubungkan terlebih dahulu.

### 2.2.2.3 Arsitektur Sistem

Arsitektur sistem merujuk pada konfigurasi secara keseluruhan dari Sistem Manajemen Basis Data, basis data dan aplikasi yang memanfaatkannya. Beberapa jenis arsitektur sistem yang dapat digunakan akan diuraikan sebagai berikut [FAT-02:150]:

#### 2.2.2.3.1 Sistem Tunggal (*Stand-Alone*)

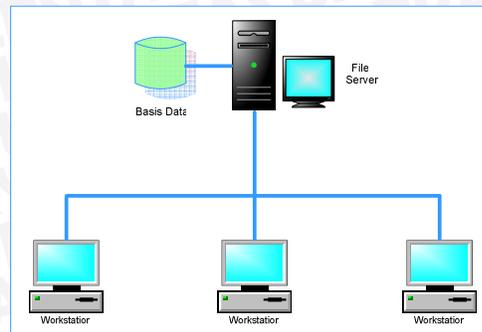
Sistem Manajemen Basis Data, basis data, dan aplikasi basis data ditempatkan pada komputer yang sama pada arsitektur sistem tunggal. Arsitektur ini digunakan jika basis data yang dikelola tidak terlalu besar dan bersifat membantu pekerjaan administratif yang ditunjukkan dalam Gambar 2.6 [FAT-02:150].



**Gambar 2.6** Sistem Tunggal (*Stand-Alone*)  
Sumber: [FAT-02:151]

#### 2.2.2.3.2 Sistem Terpusat (*Centralized System*)

Pada arsitektur sistem terpusat, lokasi basis data secara fisik berada pada komputer pusat dalam suatu lingkungan tertentu. Proses pengolahan data hanya berlangsung di komputer pusat meskipun pemasukan dan akses data dapat dilakukan dari berbagai terminal yang terhubung dengan komputer pusat [DHA-02:112].



**Gambar 2.7** Basis Data Terpusat  
Sumber: [FAT-02:152]

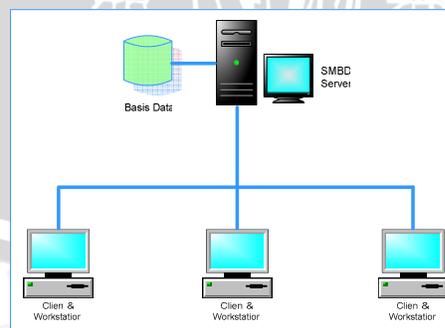
### 2.2.2.3.3 Sistem *Client-Server*

Sistem *client-server* terdiri atas dua bagian utama, yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi SMBD dan basis data. Setiap aktivitas yang dikehendaki oleh pengguna akan ditangani dahulu oleh *client*, selanjutnya *client* mengupayakan agar semua proses ditangani sendiri. *Client* akan meminta data ke *server* jika ada permintaan pada basis data oleh pengguna, contohnya untuk proses *query*.

Sistem *Client-Server* dapat dipisahkan berdasarkan kompleksitasnya menjadi [FAT-02:150]:

#### 1. Sistem *Client-Server* sederhana (Arsitektur *Two-tier*)

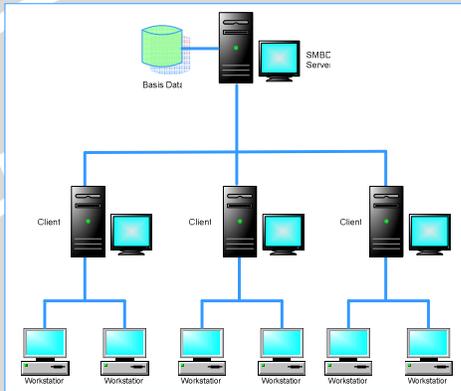
Arsitektur *Two-tier* diterapkan pada LAN (*Local Area Network*), *client* berfungsi untuk menangani sebagian besar proses pengolahan dan *workstation* berfungsi untuk menangani interaksi dengan pengguna, menerima data masukan, dan menampilkan hasil pengolahan). *Client* dan *server* yang terpisah meningkatkan kompleksitas dalam membangun aplikasi secara keseluruhan dan aplikasi *client* akan berubah bila ada perubahan pada basis data.



**Gambar 2.8** Sistem *Client-Server* Sederhana (*Two-tier*)  
Sumber: [FAT-02:154]

## 2. Sistem *Client-Server* kompleks (Arsitektur *Three-tier*)

Arsitektur *Three-tier* atau arsitektur *Multi-tier* (*N-Tier*) menempatkan aplikasi basis data pada setiap *client*. *Client* ditempatkan pada lokasi yang lebih dekat dengan kumpulan *workstation*. Setiap *client* dan sejumlah *workstation* membentuk sebuah LAN sendiri. *Client* dapat dipandang sebagai *server* aplikasi karena menjadi tempat penyimpanan basis data dan turut menangani proses-proses dalam aplikasi.



**Gambar 2.9** Sistem *Client-Server* Kompleks (*N-tier*)  
Sumber: [FAT-02:155]

### 2.2.3 *Structure Query Language* (SQL)

Sejarah SQL dimulai dari artikel seorang peneliti dari IBM bernama Dr. Edgar F. Codd yang membahas tentang ide pembuatan basis data relasional pada bulan Juni tahun 1970. Penelitian di atas juga membahas tentang kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa standar tersebut kemudian diberi nama SEQUEL (*Structured English Query Language*).

IBM (*International Business Machine*) mengadakan proyek pembuatan basis data relasional bernama SEQUEL. IBM mengubah penamaan SEQUEL menjadi SQL karena permasalahan hukum. Implementasi basis data relasional dikenal dengan System/R.

Perusahaan bernama Oracle yang membuat *server* basis data populer yang bernama sama dengan nama perusahaannya dibangun pada akhir tahun 1970. Dengan meningkatnya kepopuleran Oracle, SQL juga ikut populer sehingga saat ini menjadi standar *defacto* bahasa dalam manajemen basis data.

Standarisasi SQL dimulai pada tahun 1986, ditandai dengan dikeluarkannya standar SQL oleh ANSI. Standar ini sering disebut dengan SQL86. Standar tersebut kemudian diperbaiki pada tahun 1989 kemudian diperbaiki lagi pada tahun 1992, versi

terakhir dikenal dengan SQL92. Pada tahun 1999 dikeluarkan standar baru yaitu SQL99, akan tetapi kebanyakan implementasi basis data mereferensi pada SQL92 [FAT-02:220].

SQL adalah sebuah bahasa basis data standar, sebuah bahasa dengan standar industri untuk membuat, meng-*update*, dan melakukan *query* pada sistem manajemen basis data relasional [PIN-05:56].

### 2.2.3.1 Data Definition Language (DDL)

*Data Definition Language* merupakan kelompok perintah SQL yang digunakan untuk mendefinisikan objek dalam basis data. Objek tersebut dapat berupa *database*, *table*, *key*, atau *field*. Contoh perintah SQL dalam kelompok DDL adalah [PIN-05:57]:

- CREATE TABLE – membuat sebuah tabel *database* baru.
- ALTER TABLE – meng-*alter* (mengubah) sebuah tabel *database*.
- DROP TABLE – menghapus sebuah tabel *database*.
- CREATE INDEX – membuat sebuah indeks (kunci pencarian).
- DROP INDEX – menghapus sebuah indeks.

### 2.2.3.2 Data Manipulation Language (DML)

*Data Manipulation Language* merupakan kelompok perintah SQL yang digunakan untuk memanipulasi data/*record* pada basis data. Contoh perintah SQL yang termasuk dalam kelompok DML adalah [PIN-05:57]:

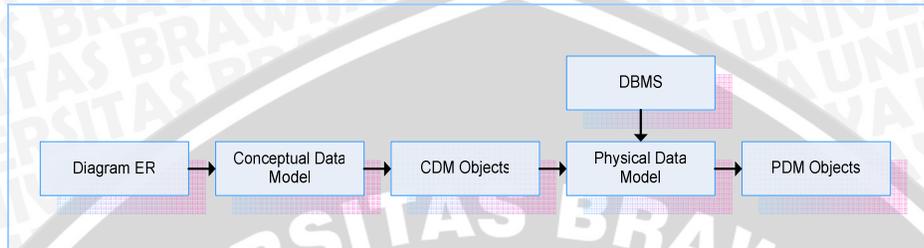
- SELECT – mengekstrak data dari sebuah tabel *database*.
- UPDATE – meng-*update* data dari sebuah tabel *database*.
- DELETE – menghapus data dari sebuah tabel *database*.
- INSERT INTO – memasukkan data baru kedalam sebuah tabel *database*.

### 2.2.4 Sybase Power Designer 10

Sybase Power Designer 10 adalah perangkat lunak untuk merancang basis data. Sybase Power Designer 10 merancang basis data dengan cara mendesain diagram *Entity-Relationship*, menganalisis diagram *Entity-Relationship*, dan mengubah diagram *Entity-Relationship* menjadi sebuah basis data secara fisik. Perangkat lunak Power Designer 10 terdiri dari 6 model, yaitu *Business Proses Model*, *Conceptual Data Model*, *Free Model*, *Object-Oriented Model*, *Physical Data Model* dan *XML Model* [POW-04].

*Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) digunakan pada pembuatan dan penganalisaan diagram *Entity Relationship* untuk basis data sistem informasi kehadiran mahasiswa.

Konsep kerja Sybase Power Designer 10 ditunjukkan dalam Gambar 2.11 berikut:



**Gambar 2.10** Konsep kerja Sybase Power Designer 10  
Sumber: [POW-04]

Proses yang terjadi pada Power Designer 10 dijelaskan dalam Gambar 2.11, yaitu masukan, proses dan keluaran pada *Conceptual Data Model* dan *Physical Data Model*. Proses yang terjadi pada *Conceptual Data Model* dan *Physical Data Model* akan dijelaskan sebagai berikut:

#### 2.2.4.1 *Conceptual Data Model* (CDM)

*Conceptual Data Model* adalah model data yang merepresentasikan struktur logika secara konseptual dari suatu basis data. Model konseptual ini terdiri dari beberapa objek yang akan digunakan untuk membentuk suatu basis data. *Conceptual Data Model* tidak terkait dengan Sistem Manajemen Basis Data sehingga pada perancangan diagram *Entity-Relationship* tidak perlu dipertimbangkan bentuk fisik dari basis data tersebut [POW-04].

Langkah-langkah dalam pembuatan *Conceptual Data Model* pada Power Designer 10 adalah:

- Menggambarkan objek-objek atau atribut-atribut dari data yang akan dibuat disertai dengan hubungan antar atribut dalam format grafik.
- Memeriksa kebenaran dari desain diagram *Entity Relationship* tersebut, dengan menggunakan *tool Check Model* pada *toolbar* Power Designer 10.

- Mengubah model *Conceptual Data Model* yang sudah benar menjadi model *Physical Data Model* (*generate*), dengan menggunakan *tool Generate Physical Data Model* pada *toolbar* Power Designer 10. Model fisik akan terbentuk melalui *Physical Data Model* dari basis data yang akan dibuat.

#### 2.2.4.2 *Physical Data Model* (PDM)

*Physical Data Model* merupakan bentuk implementasi fisik dari basis data. *Physical Data Model* merepresentasikan basis data lebih nyata daripada *Conceptual Data Model*. Perubahan struktur yang terjadi dari model *Conceptual Data Model* menjadi model *Physical Data Model* ditunjukkan dalam Tabel 2.2 berikut ini [POW-04]:

**Tabel 2.2** Perubahan struktur CDM ke PDM

Objek pada CDM	Setelah di-generate ke PDM
<i>Entity</i>	Tabel
Atribut pada <i>entity</i>	Kolom pada tabel
<i>Primary Identifier</i>	<i>Primary</i> atau <i>Foreign Key</i> (berdasarkan <i>independent</i> atau <i>dependent relationship</i> )
<i>Identifier</i>	<i>Primary</i> atau <i>Foreign Key</i>
<i>Relationship</i>	<i>Reference</i>

Sumber : [POW-04]

Langkah-langkah yang dilakukan pada model *Physical Data Model*, yaitu:

- Memeriksa kebenaran model *Physical Data Model* setelah di-generate, dengan menggunakan *tool Check Model* pada *toolbar* Power Designer 10.
- Meng-generate model tersebut menjadi basis data yang nyata sesuai dengan Sistem Manajemen Basis Data yang diinginkan. Sistem basis data yang akan digunakan harus sudah tercantum pada *Open Database Connectivity Data Source*.
- Menyimpan *query* yang telah dibentuk ke dalam suatu direktori dan memeriksa kebenaran *query* tersebut.
- Melakukan koneksi dengan *Open Database Connectivity* yang diinginkan dan membuat basis data sesuai dengan *query* yang telah diperiksa kebenarannya. Basis data yang telah terbentuk dapat secara langsung digunakan atau dimanipulasi.

### 2.2.5 Borland Interbase

Borland dikenal dengan produksi aplikasi-aplikasi pengembangan seperti Delphi, Jbuilder, C++ Builder, juga produk *server* seperti Interbase, Jdatastore, Borland Enterprise Server serta masih banyak lagi produk-produk dari Borland.

Interbase mendukung arsitektur aplikasi *client-server*, yaitu basis data tidak tersebar di setiap komputer *client*. *Server* berada terpusat dan setiap *client* dapat mengakses *server* setiap saat menggunakan *interface* tertentu melalui jaringan yang sudah ada [DWI-03:1-2].

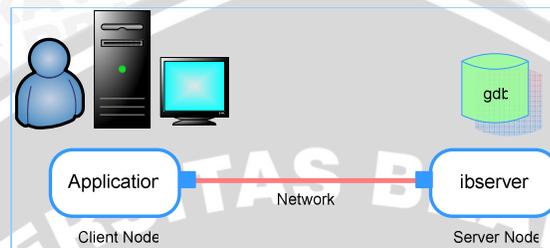
#### 2.2.5.1 Hal Utama Interbase Server

Interbase mempunyai hal-hal yang menjadi keutamaan yaitu [DWI-03:3]:

- *Network Protokol Support*: Interbase dapat berjalan di semua platform protokol jaringan seperti TCP/IP, NetBEUI/*named pipes*, Netware IPX/SPX.
- Menggunakan SQL-92 entry-level: ANSI Standard SQL, tersedia interaktif SQL *tool* dan borland dekstop aplikasi.
- *Simultan access* ke multi *database*: beberapa aplikasi dapat mengakses beberapa *database* dalam waktu yang bersamaan.
- *Query* optimasi: *server* mengoptimasi query secara otomatis, atau dapat dibuat sendiri secara manual.
- Tipe data blob dan filter blob: ukuran karakter yang berubah-ubah, teks ataupun gambar dapat disimpan dengan mudah oleh Interbase.
- Windows ISQL: tool untuk windows digunakan untuk interaktif pendefinisian data dan query.
- *Server manager: tool* di Interbase untuk windows yang digunakan untuk *backup database, restore*, pemeliharaan dan keamanannya.
- *Triggers*: modul ini akan aktif apabila data terjadi perubahan, yaitu *insert, update* dan *delete*.
- *Updatable views*: Interbase dapat membuat *view* dari beberapa tabel dan apabila jumlah tabel berubah, maka *view* pun akan berubah. Interbase juga dapat meng-update dari *view* yang ada.
- *Outer Joins*: relational antara dua tabel atau lebih untuk operasi yang kompleks.
- *Concurrent multiple application access to data*: *client* yang membaca isi tabel tidak dapat diblok oleh yang lain.

### 2.2.5.2 Konsep *Client-Server* di Interbase

Konsep *client-server* di Interbase adalah konsep sistem yang berperan sebagai pelayan adalah *server*, dalam hal ini Interbase. *Client* dapat menggunakan berbagai aplikasi untuk meminta data yang terpisah antara Interbase *server* dengan *client* (dalam komputer yang berbeda). Satu komputer yang memenuhi kriteria *server* digunakan sebagai *database server*, sedangkan yang lain berperan sebagai *client* [DWI-03:5].



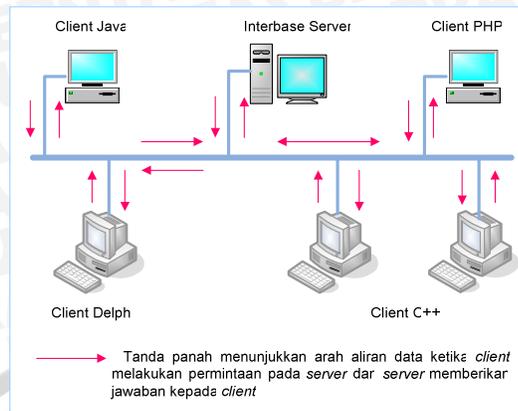
**Gambar 2.11** Arsitektur *Client-Server* pada Borland Interbase  
Sumber: [ANO-01]

#### 2.2.5.2.1 *Client*

*Client* adalah sebuah aplikasi yang dijalankan tersendiri seperti Delphi, C++ Builder, Java, PHP, Microsoft Access. *Software* tersebut dapat mengakses data di Interbase *server*. Pengaksesan data di Interbase *server* dapat dilakukan pada komputer atau *remote access* pada sebuah jaringan. *Client* mengakses data dari Interbase *server* menggunakan Interbase *client library*, yang merupakan bawaan dari Interbase. *Client* selalu melakukan permintaan data kepada *server* sehingga tanggung jawab pengolahan data terletak pada Interbase *server*.

Aplikasi *client* bertanggung jawab penuh untuk menampilkan data-data yang diinginkan oleh pengguna, sehingga beban yang ditanggung oleh aplikasi *client* pun cukup berat.

Siklus mengalirnya data saat *client* melakukan akses ke *server* dan saat *server* melakukan pelayanan memberikan data pada setiap *client* yang menggunakan aplikasi yang berbeda ditunjukkan dalam Gambar 2.13 [DWI-03:6].



**Gambar 2.12** Konsep *Client-Server*

Sumber: [DWI-03:7]

### 2.2.5.2.2 Server

*Server* melakukan pelayanan pada saat *client* melakukan proses permintaan seperti pencarian data dengan kriteria tertentu, mengurutkan data, memasukkan data dalam *database server*, mengubah data, mengeksekusi prosedur, menghapus data dan masih banyak lagi kebutuhan *client*. *Interbase server* secara otomatis akan melayani permintaan tersebut.

Konsep *client-server* pada sistem operasi umumnya sama, di mana peranan utama suatu program *server* adalah melayani *client* yang jumlahnya banyak yang memiliki tujuan untuk menggunakan secara bersama sumber daya yang ada di *server*.

Sistem operasi *server* bertugas menanti permintaan *client*. Program *server* menghabiskan sebagian besar waktu kerjanya pasif menanti permintaan *client*, melaksanakan banyak permintaan pelayanan pada saat bersamaan, serta mendahulukan *client* yang memiliki prioritas lebih tinggi. *Server* harus menyediakan beberapa tingkatan prioritas untuk *client*-nya. Oleh karena itu program *server* membutuhkan memori dan prosesor yang besar [DWI-03:6-7].

## 2.3 Borland Delphi

Delphi berasal dari bahasa pemrograman yang cukup terkenal, yaitu Pascal. Bahasa Pascal sendiri telah diciptakan pada tahun 1971 oleh ilmuwan dari Swiss, yaitu Niklaus Wirth. Nama Pascal diambil dari ahli matematika dan filsafat dari Perancis, yaitu Blaise Pascal.

Turbo Pascal yang dirilis oleh Borland International Incorporation tahun 1983. Turbo Pascal ini memiliki sedikit perbedaan dengan Pascal standar, antara lain dalam hal pengolahan string, penambahan beberapa prosedur dan fungsi.

Turbo Pascal yang muncul pertama kali hanya dapat dijalankan di sistem operasi DOS. Namun dalam perkembangan selanjutnya, Borland International juga merilis Turbo Pascal yang berjalan di Windows 3.x, yaitu Turbo Pascal For Windows.

Pada tahun 1992, Borland International menggabungkan Turbo Pascal For DOS dan Turbo Pascal For Windows menjadi satu paket bahasa pemrograman yang dikenal dengan nama Borland Pascal versi 7.

Borland International mengembangkan bahasa Pascal yang bersifat visual sejak tahun 1993 karena pemrograman Windows dengan Borland Pascal masih dirasa cukup sulit. Hasil dari pengembangan ini adalah dirilisnya Delphi 1 pada tahun 1995.

Borland International merilis Delphi 2 pada tahun 1996 yang sudah bersifat 32 bit. Delphi 2 hanya bisa dijalankan pada Windows 95 dan Windows NT.

Borland International yang berganti nama menjadi Inprise Corporation berturut-turut kembali merilis penyempurnaan Delphi, yaitu Delphi 3, 4, dan 5 pada tahun 1997, 1998, dan 1999 [PRA-00].

Delphi adalah kompiler atau penerjemah bahasa Delphi yang awalnya dari bahasa Pascal yang merupakan bahasa tingkat tinggi sekelas dengan Basic. Bahasa pemrograman di Delphi disebut bahasa prosedural artinya bahasanya mengikuti urutan tertentu. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa prolog. Delphi termasuk keluarga visual sekelas Visual Basic dan Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram dapat memilih objek apa yang ingin dimasukkan ke dalam *form*, lalu tingkah laku objek tersebut saat menerima aksi (*event*) tinggal dibuat programnya. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti, dan metode atau prosedur dikemas menjadi satu kemasan (*encapsulate*) [MAR-03].

Sesuai namanya, pada pemrograman berorientasi objek, segala sesuatu yang dimanipulasi dalam program adalah objek. Setiap objek selalu mempunyai dua elemen, yaitu data dan metode. Pada dunia pemrograman, data dapat dianalogikan dengan variabel sedangkan metode dapat dianalogikan dengan fungsi. Setiap bahasa pemrograman berorientasi objek harus mendukung tiga konsep dasar berikut [PRA-00:91-94]:

1. *Encapsulation*

Konsep pengkapsulan yaitu penggabungan data dan metode ke dalam satu kesatuan yang disebut kelas.

## 2. Inheritance

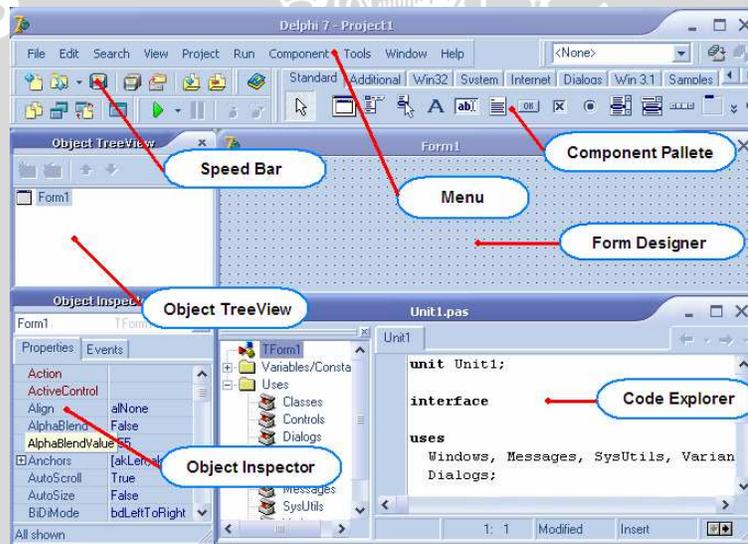
Konsep pewarisan memungkinkan untuk membuat kelas baru yang memiliki data dan metode dari kelas lain tetapi juga memiliki data dan metode sendiri.

## 3. Polymorphism

Bila diterjemahkan secara harfiah, polimorfisme berarti banyak bentuk. Pada pemrograman berorientasi objek, polimorfisme dapat diartikan sebagai dua kelas yang diturunkan dari satu kelas mempunyai metode yang sama tetapi implementasinya berbeda.

### 2.3.1 Lingkungan IDE (*Integrated Development Environment*) Delphi

Delphi memiliki Lingkungan IDE yang digunakan sebagai tempat untuk membangun perangkat lunak. Lingkungan IDE menyediakan perangkat-perangkat yang digunakan untuk membangun suatu perangkat lunak dengan cepat dan mudah. Lingkungan IDE pada Delphi ditunjukkan dalam Gambar 2.13 [PRA-04:4-8].



Gambar 2.13 Tampilan IDE Delphi  
Sumber: [PRA-02:4]

IDE Delphi dibagi menjadi:

- *Menu*

*Menu* pada Delphi memiliki kegunaan seperti *menu* pada aplikasi Windows.

- *Speed Bar*

*Speed bar* berisi kumpulan tombol pengganti beberapa item *menu* yang sering digunakan. Setiap tombol pada *speed bar* menggantikan salah satu item *menu*.

- *Component Palette*  
*Component palette* berisi kumpulan ikon yang melambangkan komponen-komponen pada VCL (*Visual Component Library*) atau CLX (*Component Library for Cross Platform*).
- *Form Designer*  
*Form designer* merupakan tempat untuk merancang jendela aplikasi. Perancangan form dilakukan dengan meletakkan komponen-komponen yang diambil dari *component palette*.
- *Code Explorer*  
*Code Explorer* adalah tempat menuliskan program dan meletakkan pernyataan-pernyataan dalam bahasa *Object Pascal*.
- *Object TreeView*  
*Object treeview* berisi daftar komponen yang telah diletakkan pada *form designer*.
- *Object Inspector*  
*Object inspector* digunakan untuk merubah karakteristik sebuah komponen. *object inspector* memiliki dua tabs, yaitu *properties* dan *events*.

#### 2.4 **Barcode**

*Barcode* adalah suatu metode dalam menggantikan karakter memakai susunan garis-garis vertikal dengan ketebalan tertentu berdasarkan sistem bilangan biner, sehingga *barcode* dapat secara langsung diakses dengan sistem yang dikontrol oleh komputer [IND-03:1].

*Barcode* ini terdiri atas dua unsur dasar, yaitu bar dan spasi. Bar adalah garis hitam vertikal, sedangkan spasi merupakan jalur putih vertikal antara dua bar. Bentuk *barcode* ini mempunyai tebal yang berbeda-beda. Variasi tebal tipis *barcode* inilah yang memuat informasi yang akan dibaca.

Tipe dan kegunaan dari *barcode* ada berbagai macam, tergantung pada fungsi dan penggunaannya. Sebagai contoh, *barcode* tipe Code 39 yang berguna untuk identifikasi orang atau barang.

Tipe dan kegunaan dari *barcode* ada berbagai macam, tergantung pada fungsi dan penggunaannya. Tipe *barcode* yang akan digunakan tergantung dari 3 faktor yaitu [IDA-05:1] :

- Implementasi
- Data yang akan dikodekan
- Bagaimana cara mencetak *barcode*

Ada beberapa macam tipe *barcode* standar untuk bermacam-macam kegunaan antara lain ditunjukkan dalam Tabel 2.3.

**Tabel 2.3** Tabel Kegunaan *Barcode*

<i>Barcode</i>	Kegunaan
Data Matrix	Identifikasi unik yang digunakan oleh departemen pertahanan Amerika Serikat.
EAN	Barang yang diperjualbelikan ke seluruh dunia.
Code 128	Identifikasi untuk perdagangan global, biasa terdapat pada karton pembungkus barang.
EAN-13	Identifikasi buku
Code 39	Identifikasi pengenalan identitas orang atau barang
POSTNET	Digunakan pada kantor Pos Amerika Serikat
UPC	Barang yang diperjualbelikan ke seluruh dunia.

Sumber: [IDA-05:1]

#### 2.4.1 EAN (*European Article Numbering*)

EAN merupakan pengembangan dari UPC, yang diperkenalkan pertama kali pada tahun 1977 oleh lembaga EAN. EAN digunakan sebagai kode pengenal bagi barang yang diperjualbelikan ke seluruh dunia.

EAN mengkodekan 13 karakter, dua atau tiga dari kode awal menunjukkan kode negara. Diikuti oleh 10 kode yang menunjukkan kode data dan kode pengenal.



**Gambar 2.14** *Barcode* tipe EAN

Sumber: [IND-03]

Keuntungan dari penggunaan EAN antara lain:

1. Tidak perlu memberi label harga pada semua barang yang digunakan. Hal ini dikarenakan semua harga barang disimpan pada satu basis data yang dapat diakses melalui nomer pada *barcode*.
2. Cepat dan aman dan minimal dalam kesalahan.

### 2.4.2 Code 128

Code 128 dikenalkan pada tahun 1981, sebagai pengembangan dari EAN yang dapat mengenali karakter *alphanumeric* maupun karakter numerik.

Code 128 adalah *barcode* dengan kerapatan tinggi, dapat mengkodekan keseluruhan simbol ASCII (128 karakter) dalam luasan yang paling minimal dibandingkan dengan *barcode* jenis lain, hal ini disebabkan karena code 128 menggunakan 4 ketebalan elemen (bar atau spasi) yang berbeda (jenis yang lain menggunakan 2 ketebalan elemen yang berbeda).



Gambar 2.15 Barcode tipe Code 128  
Sumber: [IND-03]

Setiap karakter pada code 128 dikodekan oleh 3 bar dan 3 spasi (atau 6 elemen) dengan ketebalan masing-masing elemen 1 sampai 4 kali ketebalan minimum (*module*), jika dihitung dengan satuan *module* maka tiap karakter code 128 terdiri dari 11 *module* kecuali untuk *stop character* yang terdiri dari 4 bar 3 spasi (13 *module*). Jumlah total *module* untuk bar selalu genap sedangkan untuk spasi selalu ganjil, selain itu code 128 memiliki 3 *start character* yang berbeda sehingga code 128 memiliki 3 sub set karakter yang bersesuaian dengan start karakternya

### 2.4.3 Code 93

Code 93 diperkenalkan pada tahun 1982 sebagai pengembangan dari code 39 dengan kerapatannya yang lebih tinggi. Karakter dari tiap code 93 memiliki 9 *module* baik hitam maupun putih. Tiap karakter mempunyai 3 garis dan 3 spasi, dimana tiap karakternya dimulai dengan garis dan diakhiri dengan spasi. Sebagai karakter yang menunjukkan kode awal dan kode akhir digunakan simbol " ? ". Seperti halnya code 39, code 93 dapat mengkodekan karakter *alphanumeric* yaitu angka desimal dan huruf besar serta tambahan karakter spesial *-.\*/%+*.



Gambar 2.16 Barcode tipe Code 93  
Sumber: [IND-03]

#### 2.4.4 Code 39

Code 39 adalah *barcode* yang dikembangkan oleh Intermec pada tahun 1969. Code 39 merupakan kode diskrit dengan dua lebar bar (garis vertikal hitam) yang berbeda, dalam arti tiap bar mengkodekan satu karakter dengan menggunakan spasi (garis vertikal putih) yang telah ditentukan lebarnya. [IND-03:1]. Code 39 dapat mengkodekan karakter *alphanumeric* yaitu angka desimal dan huruf besar serta tambahan karakter spesial  $-\cdot*\$/\%+$ .

Keunggulan dari code 39 antara lain [IDA-05:1]:

1. Mudah dalam penggunaannya terutama dalam mengkodekan karakter *alphanumeric*.
2. Mempunyai karakter khusus sebagai awalan dan akhiran, sehingga tidak ada data yang terpisah-pisah.
3. Didesain agar dapat mengecek karakter secara otomatis.

Satu karakter dalam code 39 terdiri dari 9 elemen yaitu 5 bar dan 4 spasi yang disusun bergantian antara bar dan spasi. Elemen yang lebar mewakili digit biner 1 dan elemen yang sempit mewakili digit biner 0. Kode awal (pada kiri baris) dan akhir (pada kanan baris) pada code 39 biasanya sama dan disimbolkan dengan huruf yang telah ditentukan atau dengan menggunakan simbol tanda bintang (\*).



Gambar 2.17 Barcode tipe Code 39  
Sumber: [IND-03]

#### 2.4.5 UPC (*Universal Product Code*)

UPC pertama kali digunakan sebagai kode pengenal untuk barang dan perusahaan yang membuatnya pada tahun 1974 di Amerika Serikat dan Kanada.

UPC mengkodekan 12 digit sebagai **SLLLLLMRRRRRE**, dimana S (start) dan E (End) adalah bit dengan pola 101, M (middle) adalah bit dengan pola 01010 (dikenal sebagai *guard bars*), dan L (left) serta R (right) adalah digit dengan panjang masing-masing 7 bit. Sehingga total satu kode UPC adalah 95 bit. UPC hanya mengkodekan angka tanpa ada huruf maupun karakter.



Gambar 2.18 Barcode tipe UPC  
Sumber: [IND-03]

Tabel karakter code 39 beserta nilai karakternya tercantum dalam Tabel 2.4.

**Tabel 2.4** Tabel karakter code 39 beserta nilai karakternya

Karakter ASCII	Karakter Set									Nilai Karakter
	Digit Biner									
	B	S	B	S	B	S	B	S	B	
0	0	0	0	1	1	0	1	0	0	0
1	1	0	0	1	0	0	0	0	1	1
2	0	0	1	1	0	0	0	0	1	2
3	1	0	1	1	0	0	0	0	0	3
4	0	0	0	1	1	0	0	0	1	4
5	1	0	0	1	1	0	0	0	0	5
6	0	0	1	1	1	0	0	0	0	6
7	0	0	0	1	0	0	1	0	1	7
8	1	0	0	1	0	0	1	0	0	8
9	0	0	1	1	0	0	1	0	0	9
A	1	0	0	0	0	1	0	0	1	10
B	0	0	1	0	0	1	0	0	1	11
C	1	0	1	0	0	1	0	0	0	12
D	0	0	0	0	1	1	0	0	1	13
E	1	0	0	0	1	1	0	0	0	14
F	0	0	1	0	1	1	0	0	0	15
G	0	0	0	0	0	1	1	0	1	16
H	1	0	0	0	0	1	1	0	0	17
I	0	0	1	0	0	1	1	0	0	18
J	0	0	0	0	1	1	1	0	0	19
K	1	0	0	0	0	0	0	1	1	20
L	0	0	1	0	0	0	0	1	1	21
M	1	0	1	0	0	0	0	1	0	22
N	0	0	0	0	1	0	0	1	1	23
O	1	0	0	0	1	0	0	1	0	24
P	0	0	1	0	1	0	0	1	0	25
Q	0	0	0	0	0	0	1	1	1	26
R	1	0	0	0	0	0	1	1	0	27
S	0	0	1	0	0	0	1	1	0	28
T	0	0	0	0	1	0	1	1	0	29
U	1	1	0	0	0	0	0	0	1	30
V	0	1	1	0	0	0	0	0	1	31
W	1	1	1	0	0	0	0	0	0	32
X	0	1	0	0	1	0	0	0	1	33
Y	1	1	0	0	1	0	0	0	0	34
Z	0	1	1	0	1	0	0	0	0	35
-	0	1	0	0	0	0	1	0	1	36
.	1	1	0	0	0	0	1	0	0	37
Spasi	0	1	1	0	0	0	1	0	0	38
*	0	1	0	0	1	0	1	0	0	-
\$	0	1	0	1	0	1	0	0	0	39
/	0	1	0	1	0	0	0	1	0	40
+	0	1	0	0	0	1	0	1	0	41
%	0	0	0	1	0	1	0	1	0	42

Sumber: [IND-03:1]

## 2.5 Jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan hardware/software yang terhubung dengan jaringan. Setiap komputer, printer atau periferal yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node* [YUH-03].

### 2.5.1 Jenis-jenis Jaringan Komputer

Secara geografis jaringan komputer dibagi menjadi 4 (empat) jenis, yaitu [DHA-02:63]:

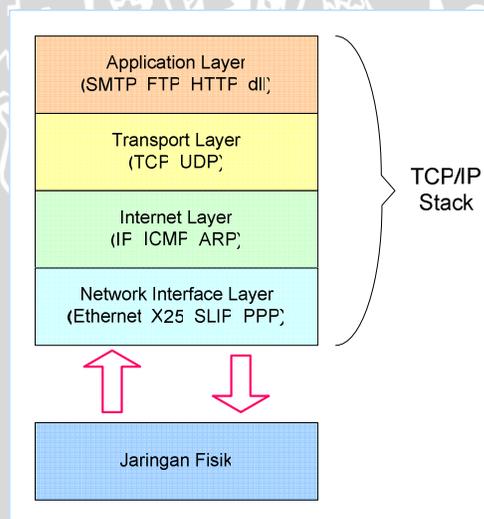
- *Local Area Network*  
*Local Area Network* merupakan jaringan dalam skala kecil sampai menengah. *Local Area Network* dapat digunakan pada gedung perkantoran atau kampus. *Local Area Network* digunakan untuk menghubungkan komputer-komputer sebagai sarana untuk memakai sumber daya secara bersama-sama maupun untuk bertukar informasi.
- *Metropolitan Area Network*  
*Metropolitan Area Network* merupakan perluasan dari penggunaan *Local Area Network*. *Metropolitan Area Network* dapat meliputi kantor-kantor atau gedung-gedung yang letaknya berdekatan maupun antar kota.
- *Wide Area Network*  
*Wide Area Network* jangkauannya meliputi daerah geografis yang luas, biasanya mencakup sebuah negara bahkan benua.
- *Internet*  
*Internet* merupakan kumpulan dari jaringan-jaringan komputer yang terinterkoneksi. Sehingga penggunaan *internet* memungkinkan pengguna komputer untuk melakukan hubungan, seperti bertukar informasi pada jaringan yang berbeda.

### 2.5.2 Model Referensi TCP/IP (*Transfer Control Protocol/Internet Protocol*)

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data *Wide Area Network* (WAN). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung-jawab atas bagian-bagian tertentu dari komunikasi data. Tugas masing-masing protokol menjadi jelas dan sederhana karena prinsip ini. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang masih bisa saling mengirim dan menerima data.

Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan tertentu. TCP/IP dapat berjalan di atas *interface* tertentu dengan melakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan saja.

Sekumpulan protokol TCP/IP ini dimodelkan dengan empat *layer* TCP/IP, sebagaimana ditunjukkan dalam Gambar 2.19.



**Gambar 2.19** Layer TCP/IP  
Sumber: [PUR-01:23]

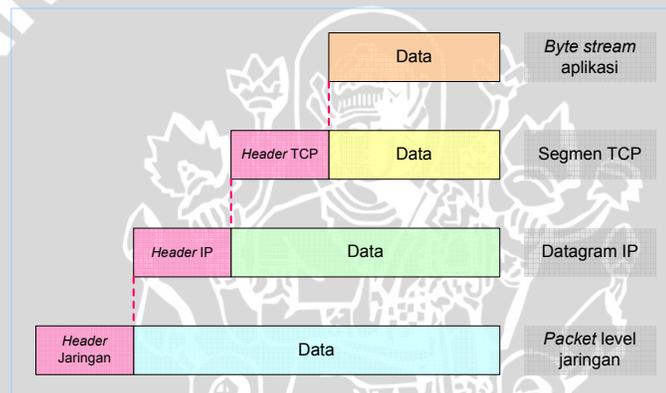
TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Keempat lapis/*layer* tersebut adalah:

- *Network Interface Layer*
- *Internet Layer*
- *Transport Layer*
- *Application Layer*

Dalam TCP/IP, terjadi penyampaian data dari protokol yang berada di satu *layer* ke protokol yang berada di *layer* yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data.

Jika suatu protokol menerima data dari protokol lain di *layer* atasnya, maka protokol penerima data akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data akan diteruskan ke protokol pada *layer* di bawahnya.

Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* di bawahnya. Jika data valid, protokol akan melepas informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang berada pada *layer* di atasnya.



**Gambar 2.20** Unit-unit Data Protocol pada Arsitektur TCP/IP  
Sumber: [STA-01:56]

Lapisan/*Layer* terbawah, yaitu *Network Interface Layer*, bertanggung-jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serat optik, atau gelombang radio. Karena tugasnya ini, protokol pada *layer* ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.

*Internet Layer* bertanggung-jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol yaitu IP, ARP, dan ICMP.

IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menentukan alamat *hardware* dari *host*/komputer yang terletak pada *network* yang sama. Sedangkan, ICMP (*Internet Control Message Protocol*) ialah protokol yang digunakan untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.

*Transport Layer* berisi protokol yang bertanggung-jawab untuk mengadakan komunikasi antara dua *host*/komputer. Kedua protokol tersebut adalah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*) [PUR-01:22-25].

TCP (*Transmission Control Protocol*) merupakan protokol yang menyediakan *service* yang dikenal sebagai *connection oriented, reliable, byte stream service*. *Connection oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan (*handshake*) terlebih dahulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan retransmisi. *Byte stream service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan [PUR-01:51].

UDP (*User Datagram Protocol*) merupakan protokol transport yang sederhana. Berbeda dengan TCP yang *connection oriented*, UDP bersifat *connectionless*. Dalam UDP tidak ada pengurutan kembali (*sequencing*) paket yang datang, *acknowledgement* terhadap paket yang datang, atau retransmisi jika paket mengalami masalah dalam pengiriman [PUR-01:55].

Pada *application layer* terletak semua aplikasi yang menggunakan protokol TCP/IP [PUR-01:25].

### 2.5.3 Topologi Jaringan Komputer

Topologi adalah cara untuk menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk sebuah jaringan. Setiap topologi yang digunakan untuk membentuk suatu jaringan mempunyai kelebihan dan kekurangan masing-masing. Beberapa jenis topologi antara lain [DHA-02:67]:

#### 1. Topologi *Bus*

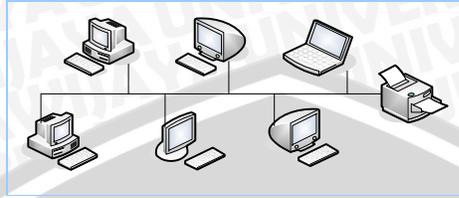
Topologi *Bus* menggunakan media kabel *coaxial* sebagai media transmisi, seperti digambarkan dalam Gambar 2.10. Keuntungan dari penggunaan topologi *bus* adalah:

- Hemat kabel.
- *Layout* kabel sederhana.
- Mudah dikembangkan.

Sedangkan kekurangan penggunaan topologi *bus* adalah:

- Kesulitan untuk mendeteksi kerusakan, apabila terjadi kerusakan dalam jaringan
- Kepadatan lalu lintas

- Seluruh jaringan akan tidak berfungsi, jika terjadi kerusakan pada kebel utama
- Diperlukan *repeater* untuk jarak jauh.



**Gambar 2.21** Topologi Bus  
Sumber: [DHA-02:67]

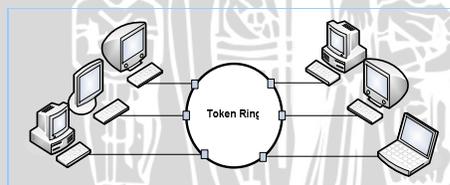
## 2. Topologi *Ring*

Topologi *ring* adalah cara menghubungkan komputer sehingga membentuk lingkaran seperti dalam Gambar 2.11. Setiap informasi yang diperoleh diperiksa alamatnya oleh *terminal* yang dilewatinya. Jika informasi yang dikirim bukan untuk terminal tersebut, informasi akan dilewatkan sampai menemukan alamat tujuan. Keuntungan penggunaan *topologi ring* adalah:

- Mudah untuk mencari kerusakan dalam jaringan.
- Tidak terdapat *collision*.

Sedangkan kekurangan dari penggunaan *topologi ring* adalah:

- Apabila satu komputer terganggu, komputer lainnya akan terganggu.
- Pengembangan jaringan lebih kaku.

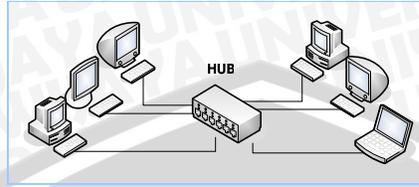


**Gambar 2.22** Topologi Ring  
Sumber: [DHA-02:68]

## 3. Topologi *Star*

Pada topologi *star*, setiap komputer dihubungkan secara langsung pada *server* atau *hub* seperti yang diperlihatkan dalam Gambar 2.12. Karena setiap komputer yang memiliki hubungan dengan *server* melalui kabel tersendiri, maka *bandwith* atau lebar jalur data dalam jaringan akan semakin lebar, sehingga penggunaan *topologi star* akan meningkatkan kinerja jaringan secara keseluruhan. Selain itu, keuntungan lain dari penggunaan topologi ini adalah apabila terdapat gangguan di suatu jalur kabel, maka jaringan tidak akan terganggu secara keseluruhan.

Dibandingkan penggunaan topologi lainnya, topologi *star* memerlukan jumlah kabel yang lebih banyak, sehingga biaya untuk menggunakan topologi ini juga akan meningkat.



**Gambar 2.23** Topologi Star  
Sumber: [DHA-02:67]

#### 2.5.4 Manfaat Jaringan Komputer

Beberapa manfaat dengan adanya jaringan komputer antara lain [TAN-00:3]:

- *Resource Sharing*

*Resource Sharing* bertujuan agar seluruh program, peralatan, khususnya data bisa digunakan oleh setiap orang yang ada pada jaringan tanpa terpengaruh oleh lokasi *resource* dan pemakai. Tujuan ini dapat diringkas dengan mengatakan bahwa *resource sharing* adalah suatu usaha untuk menghilangkan kendala jarak.

- Keandalan Tinggi

Jaringan komputer memberikan sumber-sumber alternatif yang tersedia untuk digunakan. Misalnya, semua *file* dapat disalin ke dua atau tiga buah komputer. Bila salah satu komputer tersebut tidak dapat dipakai, maka salinan yang ada pada mesin lainnya dapat digunakan. Selain itu, dengan adanya *Central Processing Unit* jamak maka bila salah satu *Central Processing Unit* tidak dapat dipakai, *Central Processing Unit* lainnya akan mengambil alih tugasnya, walaupun kinerjanya menurun.

- Menghemat Uang

Penggunaan jaringan komputer dapat menghemat pengeluaran pada suatu organisasi, hal ini disebabkan karena manfaat dari penggunaan sumber daya secara bersama-sama.

- Skalabilitas

Kemampuan untuk meningkatkan kinerja sistem secara berangsur-angsur sesuai dengan beban pekerjaan dengan hanya menambahkan sejumlah prosesor.

### BAB III

#### METODE PENELITIAN

Pada tahap ini dijelaskan mengenai langkah-langkah yang akan dilakukan untuk merealisasikan aplikasi perangkat lunak yang akan dibuat. Adapun langkah-langkahnya adalah sebagai berikut:

#### 3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung dalam perancangan dan perrealisasian aplikasi, yaitu:

- a. Perancangan sistem informasi berbasis *client-server*, meliputi :
  - Diagram Konteks
  - Data Flow Diagram Levelled
  - State-Transition Diagram
  - Entity-Relationship Diagram (ER Diagram)
- b. Basis data, meliputi :
  - Basis data SQL (*Structured Query Language*)
  - Borland Interbase
  - Sybase Power Designer 10
- c. Teori dasar pemrograman, yaitu Borland Delphi
- d. Konsep *barcode*, meliputi pengertian *barcode*, serta tipe dan kegunaannya
- e. Jaringan komputer, meliputi pengertian jaringan komputer beserta manfaatnya.

#### 3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari analisis kebutuhan perangkat lunak, perancangan sistem, perancangan basis data, perancangan proses, dan perancangan antarmuka. Perancangan didasarkan pada teori-teori yang nantinya akan diaplikasikan pada sistem. Perancangan perangkat lunak dilakukan pada tiap-tiap blok aplikasi sistem untuk mempermudah analisis terstruktur, perancangan, dan pemrograman. Blok diagram aplikasi sistem yang dibuat meliputi:

1. Aplikasi *server* yang berguna dalam mencetak *barcode*, administrasi basis data kehadiran, dan memberikan report kehadiran mahasiswa.

2. Aplikasi *client* yang digunakan untuk pengecekan *barcode* dan menyimpan informasi kehadiran mahasiswa dan mengirimkan data kehadiran mahasiswa kepada aplikasi *server*.

### 3.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak terdiri dari implementasi basis data dan implementasi antarmuka perangkat lunak. Implementasi perangkat lunak dilakukan berdasarkan pada perancangan perangkat lunak.

Implementasi basis data dilakukan pada basis data Borland Interbase. Implementasi basis data Borland Interbase terdiri dari konfigurasi basis data Borland Interbase dan implementasi perancangan basis data menggunakan SQL. Implementasi antarmuka aplikasi terdiri dari perangkat lunak untuk *server* dan perangkat lunak untuk *client*.

### 3.4 Pengujian dan Analisis Perangkat Lunak

Tahap ini adalah pengujian dan analisis yaitu menguji tiap blok sistem yang dibuat dan dibandingkan dengan teori yang ada sehingga didapatkan suatu kesimpulan mengenai aplikasi Sistem Informasi Kehadiran Mahasiswa yang dibuat.

Pengujian dilakukan terhadap basis data kehadiran dan pengujian selanjutnya adalah mengimplementasikan aplikasi Sistem Informasi Kehadiran Mahasiswa secara *client-server*. Pengujian dan analisis dilakukan terhadap permasalahan-permasalahan yang ada selama proses aplikasi dijalankan.

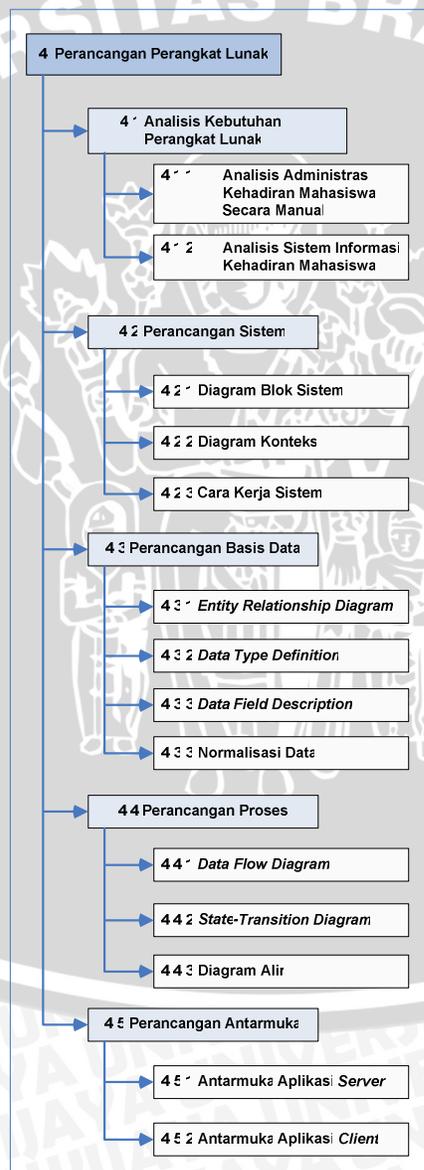
### 3.5 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dari aplikasi yang telah dibuat dilakukan setelah semua tahapan perancangan dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan ini merupakan informasi akhir dari perancangan aplikasi yang berisi mengenai berhasil atau tidaknya aplikasi tersebut dijalankan.

Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan.

## BAB IV PERANCANGAN PERANGKAT LUNAK

Bab ini menjelaskan perancangan Sistem Informasi Kehadiran Mahasiswa dengan menggunakan *barcode* berbasis pemrograman Borland Delphi dan basis data Borland Interbase. Perancangan yang dilakukan meliputi empat tahap, yaitu perancangan perangkat lunak, perancangan basis data, perancangan proses dan perancangan antarmuka pengguna. Diagram pohon perancangan perangkat lunak ditunjukkan dalam Gambar 4.1.



**Gambar 4.1** Diagram Pohon Perancangan Perangkat Lunak  
Sumber: *Perancangan*

Sistem Informasi Kehadiran Mahasiswa dirancang menggunakan arsitektur *client-server*. *Server* menyimpan data pada basis data dan *client* menerima masukan data melalui antarmuka pengguna, antara *client* dan *server* dihubungkan dengan jaringan intranet menggunakan protokol TCP/IP. Program untuk merancang antarmuka pengguna menggunakan Borland Delphi (versi 7). Program yang digunakan sebagai *server* basis data adalah Borland Interbase (versi 6.5). Sistem operasi yang digunakan dalam perancangan aplikasi adalah Windows XP Professional Version 2002 Service Pack 1.

Sistem Informasi Kehadiran Mahasiswa dirancang untuk melakukan:

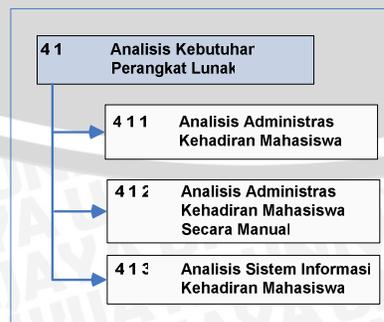
1. Penyimpanan data mahasiswa, dosen, mata kuliah, kurikulum, *login*, peserta mata kuliah, jadwal mata kuliah, dan kehadiran mahasiswa pada aplikasi *server*.
2. Pencarian data mahasiswa, dosen, mata kuliah, *login*, peserta mata kuliah, jadwal mata kuliah, dan kehadiran mahasiswa pada aplikasi *server*.
3. Pencetakan kartu kehadiran mahasiswa, kartu kehadiran dosen, dan *report* kehadiran mahasiswa pada aplikasi *server*.
4. Penyimpanan data kehadiran mahasiswa yang ditampilkan pada aplikasi *client*.

#### 4.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak adalah aktifitas rekayasa perangkat lunak yang menjembatani antara kebutuhan ditingkat sistem dan perancangan perangkat lunak [PRE-02:320].

Analisis kebutuhan adalah tahap interaksi intensif antara analisis sistem dengan komunitas pemakai sistem [IKA-05:17].

Analisis kebutuhan adalah proses yang digunakan untuk mendapatkan, menganalisis, dan memvalidasi kebutuhan-kebutuhan sistem [IAN-01:138]. Diagram pohon analisis kebutuhan perangkat lunak ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Diagram Pohon Analisis Kebutuhan Perangkat Lunak  
Sumber: Perancangan

#### 4.1.1 Analisis Administrasi Kehadiran Mahasiswa

Dalam proses analisis administrasi kehadiran mahasiswa ada beberapa tahapan yaitu menentukan komunitas pengguna (*end-user*), mengetahui keinginan pengguna, dan *entry data* bagi pengguna.

##### a) Komunitas Pengguna

Komunitas pengguna merupakan pihak yang terkait dengan proses administrasi kehadiran mahasiswa. Komunitas pengguna sistem informasi kehadiran mahasiswa sebagai berikut :

- Ketua Jurusan
- Bagian Recording
- Bagian Pengajaran
- Dosen
- Mahasiswa

##### b) Kebutuhan Pengguna

###### 1. Ketua Jurusan

Ketua jurusan membutuhkan sistem yang dapat membantu mengevaluasi jumlah kehadiran dosen dan mahasiswa pada setiap mata kuliah dalam satu semester.

###### 2. Bagian Recording

Bagian recording membutuhkan sistem yang dapat membantu proses administrasi mahasiswa, dosen, mata kuliah, kurikulum, *login*, peserta mata kuliah, jadwal kuliah, dan kehadiran mahasiswa.

###### 3. Bagian Pengajaran

Bagian pengajaran membutuhkan sistem yang dapat membantu dalam proses pengaturan jadwal kuliah dan kehadiran mahasiswa. Sistem juga diharapkan membantu pembuatan laporan kehadiran mahasiswa yang bersifat rutin dan berkala.

###### 4. Dosen

Dosen membutuhkan sistem yang dapat membantu dalam proses evaluasi kehadiran mahasiswa. Evaluasi kehadiran mahasiswa merupakan salah satu faktor yang menentukan prestasi akademik mahasiswa dalam kegiatan perkuliahan di perguruan tinggi.

#### 5. Mahasiswa

Mahasiswa membutuhkan sistem yang dapat membantu mencatat kehadiran mahasiswa secara lebih mudah dan cepat.

#### c) *Entry Data Pengguna*

Proses *entry* data pengguna dilakukan oleh pengguna yang mempunyai hak akses (wewenang) dalam sistem.

##### 1. Bagian Recording

Bagian recording memproses administrasi jadwal mata kuliah membutuhkan data dosen, mata kuliah, dan data kurikulum melalui aplikasi *server*.

##### 2. Bagian Pengajaran

Bagian pengajaran dalam memproses kehadiran mahasiswa membutuhkan data jadwal mata kuliah dan data kehadiran mahasiswa melalui aplikasi *server*.

##### 3. Dosen

Dosen melakukan *entry* data kehadirannya setelah proses otentifikasi yang mengecek data jadwal mengajar mata kuliah dan mengaktifkan kelas mengajar melalui aplikasi *client*.

##### 4. Mahasiswa

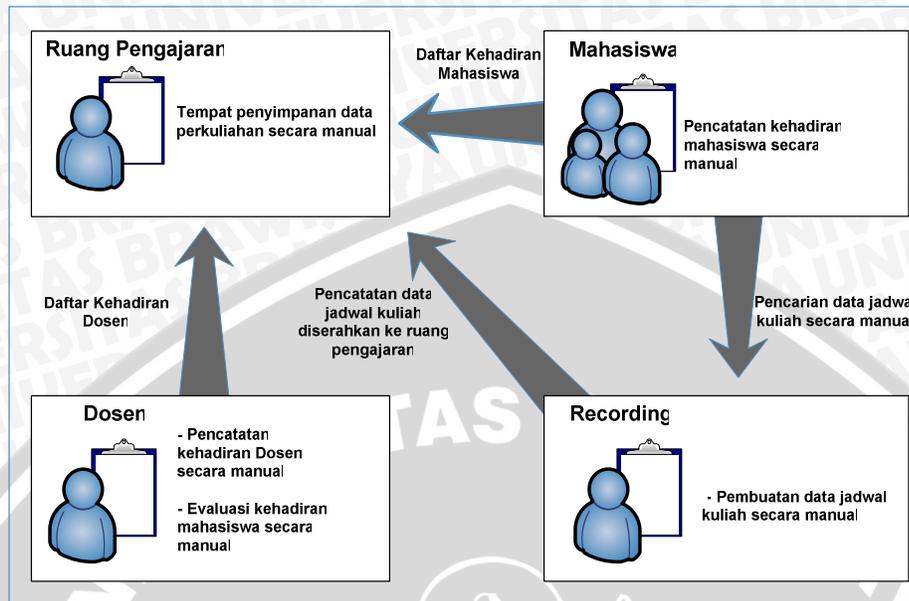
Mahasiswa melakukan proses pencatatan kehadiran setelah proses otentifikasi KRS (Kartu Rencana Studi) melalui aplikasi *client*.

#### 4.1.2 Analisis Administrasi Kehadiran Mahasiswa Secara Manual

Proses pencatatan kehadiran mahasiswa di Jurusan Teknik Elektro Universitas Brawijaya masih menggunakan cara manual, yaitu dengan membubuhkan tanda tangan setiap kali memulai satu kelas mata kuliah. Pencatatan kehadiran mahasiswa secara manual lebih sulit untuk dikelola secara efisien terutama dari penyimpanan arsip kehadiran.

Proses pencatatan kehadiran secara manual memerlukan banyak intervensi pegawai bagian administrasi maupun kejujuran mahasiswa yang sedang dicatat kehadirannya. Hal ini sering memberikan peluang adanya manipulasi data kehadiran apabila pengawasan yang kontinyu pada proses ini tidak dilakukan semestinya.

Administrasi data kehadiran secara manual ditunjukkan dalam Gambar 4.3.



**Gambar 4.3** Administrasi Kehadiran Mahasiswa Secara Manual  
Sumber: *Observasi*

Administrasi kehadiran mahasiswa secara manual memiliki kelebihan dan kekurangan, antara lain:

**A. Kelebihan**

1. Tidak memerlukan ketersediaan sumber energi listrik.
2. Tidak memerlukan ketersediaan perangkat keras (*hardware*).
3. Mudah dalam mengatasi kekurangan pengisian pencatatan kehadiran atau pemindahan jadwal mata kuliah oleh dosen pengajar.

**B. Kekurangan**

1. Proses pencatatan kehadiran secara manual membutuhkan sumber daya manusia untuk mengelola pengarsipan kehadiran.
2. Proses pencatatan kehadiran secara manual tidak bisa menjamin integritas data kehadiran secara penuh.
3. Pencarian data kehadiran mahasiswa tidak dapat dilakukan dengan mudah.

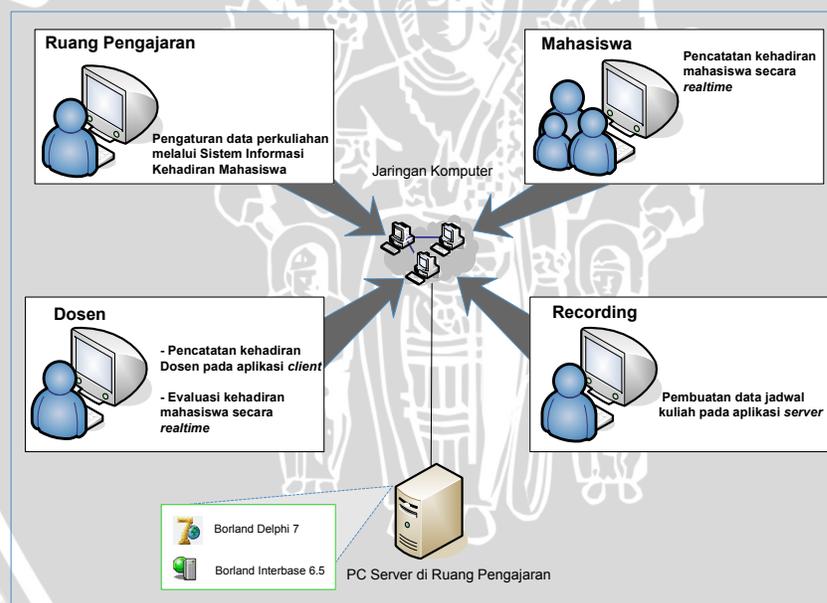
#### 4.1.3 Analisis Sistem Informasi Kehadiran Mahasiswa

Sistem Informasi Kehadiran Mahasiswa merupakan aplikasi menggunakan *barcode* memberikan manfaat berupa proses pengambilan informasi kehadiran mahasiswa menjadi lebih mudah dan cepat. Administrasi presensi mahasiswa menjadi

lebih mudah dengan bantuan aplikasi. Sistem Informasi Kehadiran Mahasiswa dibuat untuk mengatasi kekurangan yang terdapat pada pengelolaan dan pencatatan kehadiran mahasiswa secara manual.

Jadwal mata kuliah tersimpan dalam basis data untuk memudahkan dosen dalam memulai kelas mata kuliah sesuai dengan jam mengajar dosen tersebut. Kartu kehadiran dosen dibutuhkan dalam proses otentifikasi untuk mengenali dosen dan jadwal mengajar dosen yang bersangkutan pada jam mengajarnya.

Mahasiswa melakukan proses pencatatan kehadiran dengan melakukan proses otentifikasi melalui kartu kehadiran mahasiswa yang dimiliki oleh masing-masing mahasiswa. Mahasiswa diharuskan mendaftarkan mata kuliah yang akan diikuti melalui KRS dan disimpan dalam Sistem Informasi Kehadiran Mahasiswa pada tabel daftar peserta mata kuliah. Administrasi Sistem Informasi Kehadiran Mahasiswa ditunjukkan dalam Gambar 4.4.



**Gambar 4.4** Sistem Informasi Kehadiran Mahasiswa  
Sumber: Perancangan

Sistem Informasi Kehadiran Mahasiswa dirancang untuk dapat melakukan:

1. Penyimpanan data kehadiran mahasiswa, jadwal mata kuliah, dan daftar peserta mata kuliah secara terpusat dalam basis data kehadiran mahasiswa.
2. Pencetakan laporan kehadiran mahasiswa pada setiap akhir semester.

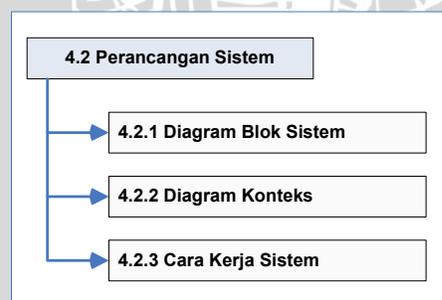
3. Pencetakan kartu kehadiran mahasiswa dan kartu kehadiran dosen untuk proses otentifikasi.
4. Pencatatan kehadiran mahasiswa dan dosen ke dalam basis data kehadiran mahasiswa.
5. Pengaturan jadwal mata kuliah yang diatur melalui aplikasi *server* oleh administrator.
6. Pencarian mahasiswa, dosen, mata kuliah, jadwal mata kuliah, daftar peserta mata kuliah, kehadiran mahasiswa secara cepat dan mudah pada aplikasi *server*.

Kekurangan yang terdapat pada Sistem Informasi Kehadiran Mahasiswa sebagai berikut:

1. Perpindahan jadwal mata kuliah diatur melalui administrator.
2. Penggunaan *barcode* sebagai otentifikasi kehadiran masih memungkinkan terjadinya kecurangan dalam proses pencatatan kehadiran.
3. Keamanan Sistem Informasi Kehadiran Mahasiswa masih terbatas pada keamanan basis data.

#### 4.2 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem meliputi diagram blok sistem, diagram konteks dan cara kerja sistem. Diagram pohon perancangan sistem ditunjukkan dalam Gambar 4.5.

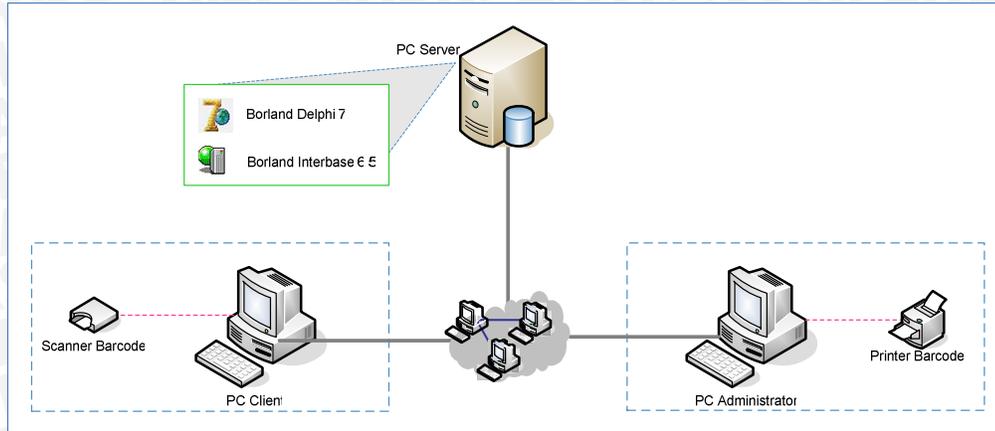


Gambar 4.5 Diagram Pohon Perancangan Sistem  
Sumber: *Perancangan*

##### 4.2.1 Diagram Blok Sistem

Diagram blok sistem menggambarkan setiap blok atau bagian dari sistem aplikasi. Sistem Informasi Kehadiran Mahasiswa dirancang untuk dijalankan pada jaringan intranet.

Diagram blok sistem aplikasi ditunjukkan dalam Gambar 4.6.



**Gambar 4.6** Diagram Blok Sistem  
Sumber: *Perancangan*

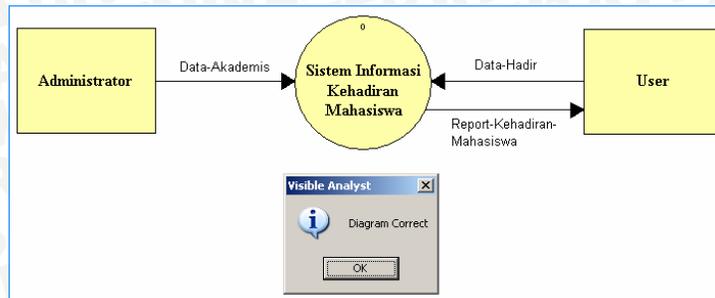
Blok diagram Sistem Informasi Kehadiran Mahasiswa meliputi:

1. Aplikasi Sistem Informasi Kehadiran Mahasiswa (SIKMA) *Server*  
Aplikasi SIKMA *Server* digunakan untuk administrasi data mahasiswa, dosen, mata kuliah, kurikulum, peserta mata kuliah, jadwal mata kuliah, kehadiran mahasiswa, *login*, pencetakan kartu kehadiran mahasiswa, pencetakan kartu kehadiran dosen, dan pencetakan *report* kehadiran mahasiswa.
2. Aplikasi Sistem Informasi Kehadiran Mahasiswa (SIKMA) *Client*  
Aplikasi SIKMA *Client* digunakan untuk mengaktifkan kelas mata kuliah oleh dosen pengajar dan menyimpan data kehadiran mahasiswa melalui pembacaan kartu kehadiran mahasiswa.

#### 4.2.2 Diagram Konteks

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem [POH-97:11].

Diagram konteks Sistem Informasi Kehadiran Mahasiswa ditunjukkan dalam Gambar 4.7.



**Gambar 4.7** Diagram Konteks  
Sumber: *Perancangan*

Diagram konteks SIKMA mempunyai komponen pembentuk yang terdiri dari administrator, *user* dan sebuah proses Sistem Informasi Kehadiran Mahasiswa.

#### 4.2.3 Cara Kerja Sistem

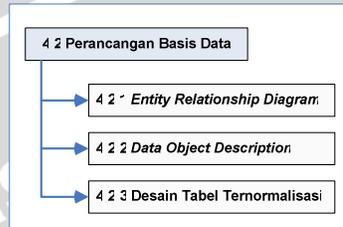
Sistem Informasi Kehadiran Mahasiswa (SIKMA) merupakan kesatuan sistem dari seluruh aplikasi yang terdapat di dalamnya. SIKMA terdiri dari aplikasi *server* dan aplikasi *client*.

Aplikasi *server* berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data mahasiswa, dosen, mata kuliah, kurikulum, *login*, peserta mata kuliah, jadwal mata kuliah, dan kehadiran mahasiswa. Kartu Kehadiran Mahasiswa (KKM), Kartu Kehadiran Dosen (KKD), dan *report* kehadiran mahasiswa dicetak melalui aplikasi *server*. Aplikasi *server* dilengkapi dengan fasilitas pencarian untuk mempermudah pekerjaan administrator dalam mengelola basis data kehadiran.

Aplikasi *client* berfungsi untuk mencatat data kehadiran mahasiswa untuk setiap pertemuan kelas mata kuliah. Aplikasi *client* diaktifkan melalui pembacaan NIP dosen pengajar berbentuk *barcode* yang dicetak dalam KKD untuk menampilkan jadwal mata kuliah serta kelas mata kuliah sesuai dengan hari dan jam mengajar dosen yang bersangkutan. Aplikasi *client* menerima masukan data dari mahasiswa berupa Nomor Induk Mahasiswa (NIM). NIM dicetak dalam bentuk *barcode* yang dinamakan Kartu Kehadiran Mahasiswa (KKM). Hasil dari pembacaan KKM oleh *barcode scanner* digunakan sebagai data masukan kehadiran mahasiswa dan disimpan dalam basis data kehadiran.

### 4.3 Perancangan Basis Data

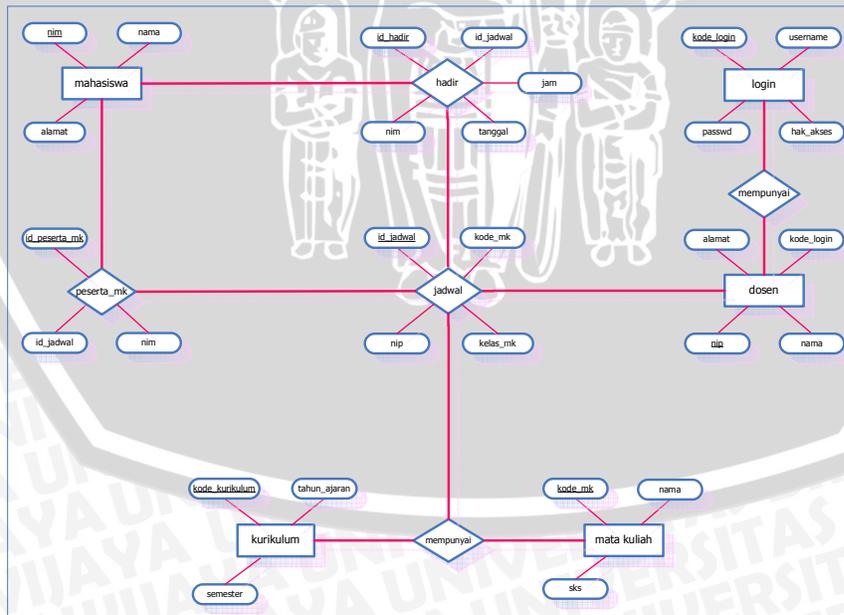
Perancangan basis data diperlukan agar basis data yang diinginkan menjadi basis data yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam manipulasi data. Perancangan basis data dapat dilakukan dengan menggunakan *entity relationship diagram*, *data object description*, dan desain basis data ternormalisasi. Diagram pohon perancangan basis data ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Diagram Pohon Perancangan Basis Data  
Sumber: Perancangan

#### 4.3.1 Entity Relationship Diagram (ERD)

*Entity relationship diagram* (ERD) digunakan untuk memodelkan struktur data dan hubungan antar data. ERD menggambarkan keadaan dunia nyata (*real world*) yang diwakili dalam bentuk entiti, atribut serta relasi. ERD memperlihatkan hubungan antara atribut yang akan dijadikan kunci untuk berelasi antar tabel. *Entity relationship diagram* dari basis data Sistem Informasi Kehadiran Mahasiswa ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Entity Relationship Diagram  
Sumber: Perancangan

### 4.3.2 Data Object Description

*Data object description* menjelaskan secara rinci mengenai atribut-atribut yang dimiliki oleh masing-masing tabel yang ada pada basis data sesuai dengan *entity relationship diagram*. *Data object description* dapat dipisahkan menjadi dua bagian, yaitu:

1. *Data type definition* menjelaskan tipe data yang digunakan oleh atribut pada masing-masing tabel yang ada di basis data kehadiran.
2. *Data field description* menjelaskan keterangan setiap atribut pada masing-masing tabel yang ada di basis data kehadiran.

#### 4.3.2.1 Data Type Definition

*Data type definition* menjelaskan secara rinci mengenai tipe data yang digunakan setiap kolom pada masing-masing tabel basis data kehadiran. Tabel-tabel yang terdapat dalam basis data kehadiran adalah:

- Tabel *m\_mhs* berisi data mahasiswa yang digunakan dalam pemilihan mata kuliah yang diprogram oleh setiap mahasiswa dan pencatatan kehadiran mahasiswa dalam setiap pertemuan mata kuliah. *Data type definition* dari tabel *m\_mhs* ditunjukkan pada Tabel 4.1.

**Tabel 4.1** Tabel *data type definition* dari tabel *m\_mhs*

Table Name	No	Field Name	Type	Key	Null	Width
m_mhs	01	nim	varchar	PK	not null	20
	02	nama	varchar			40
	03	alamat	varchar			60

- Tabel *m\_dosen* berisi data dosen pengajar yang digunakan untuk mengaktifkan mata kuliah dan kelas mata kuliah yang diajar oleh dosen mata kuliah tersebut. *Data type definition* dari tabel *m\_dosen* ditunjukkan pada Tabel 4.2.

**Tabel 4.2** Tabel *data type definition* dari tabel *m\_dosen*

Table Name	No	Field Name	Type	Key	Null	Width
m_dosen	01	nip	varchar	PK	not null	20
	02	nama	varchar			40
	03	alamat	varchar			60
	04	no_telp	varchar			20
	05	k_login	varchar	FK1	not null	20

- Tabel *m\_mk* berisi data mata kuliah yang digunakan dosen untuk mengaktifkan mata kuliah dan kelas mata kuliah.

*Data type definition* dari tabel `m_mk` ditunjukkan pada Tabel 4.3.

**Tabel 4.3** Tabel *data type definition* dari tabel `m_mk`

Table Name	No	Field Name	Type	Key	Null	Width
m_mk	01	k_mata_kuliah	varchar	PK	not null	20
	02	nama	varchar			40
	03	sks	char			1

- Tabel `m_krk` berisi data kurikulum yang digunakan sebagai acuan dalam menentukan kurikulum yang berlaku. *Data type definition* dari tabel `m_krk` ditunjukkan pada Tabel 4.4.

**Tabel 4.4** Tabel *data type definition* dari tabel `m_krk`

Table Name	No	Field Name	Type	Key	Null	Width
m_krk	01	k_krk	varchar	PK	not null	20
	02	nama	varchar			40
	03	semester	char			6
	04	krk_now	char			1

- Tabel `m_login` berisi data *login* setiap dosen yang digunakan untuk mengaktifkan mata kuliah dan kelas mata kuliah. *Data type definition* dari tabel `m_login` ditunjukkan pada Tabel 4.5.

**Tabel 4.5** Tabel *data type definition* dari tabel `m_login`

Table Name	No	Field Name	Type	Key	Null	Width
m_login	01	k_login	varchar	PK	not null	20
	02	username	varchar	unique	not null	20
	03	passwd	varchar		not null	20
	04	hak_akses	char		not null	1

- Tabel `m_hari` berisi data hari yang digunakan dalam pencatatan kehadiran mahasiswa yang disesuaikan dengan hari pada jadwal mata kuliah. *Data type definition* dari tabel `m_hari` ditunjukkan pada Tabel 4.6.

**Tabel 4.6** Tabel *data type definition* dari tabel `m_hari`

Table Name	No	Field Name	Type	Key	Null	Width
m_hari	01	k_hari	varchar	PK	not null	20
	02	nama	varchar		not null	40

- Tabel `d_peserta_mk` merupakan tabel yang berelasi dengan tabel `m_mhs` dan tabel `d_jadwal`. Tabel `d_peserta_mk` digunakan dalam pencatatan kehadiran mahasiswa sesuai mata kuliah yang diprogram oleh masing-masing mahasiswa.

Data type definition dari tabel d\_peserta\_mk ditunjukkan pada Tabel 4.7.

Tabel 4.7 Tabel data type definition dari tabel d\_peserta\_mk

Table Name	No	Field Name	Type	Key	Null	Width
d_peserta_mk	01	id_peserta_mk	smallint	PK	not null	
	02	nim	varchar	FK3	not null	20
	03	id_jadwal	smallint	FK2	not null	

- Tabel d\_jadwal merupakan tabel yang berelasi dengan tabel m\_dosen, tabel m\_mk, tabel m\_krk, dan tabel m\_hari. Tabel d\_jadwal digunakan dalam mengaktifkan mata kuliah dan kelas mata kuliah sesuai dengan login dosen pengajar. Data type definition dari tabel d\_jadwal ditunjukkan pada Tabel 4.8.

Tabel 4.8 Tabel data type definition dari tabel d\_jadwal

Table Name	No	Field Name	Type	Key	Null	Width
d_jadwal	01	id_jadwal	smallint	PK	not null	
	02	nip	varchar	FK2	not null	20
	03	k_mk	varchar	FK6	not null	20
	04	k_krk	varchar	FK7	not null	20
	05	kelas_mk	char		not null	1
	06	k_hari	varchar	FK5	not null	20
	07	jam_start	time		not null	
	08	jam_stop	time		not null	
	09	flag	char		not null	1

- Tabel d\_hadir merupakan tabel yang berelasi dengan tabel d\_jadwal dan tabel d\_peserta\_mk. Tabel d\_hadir digunakan untuk menyimpan data kehadiran mahasiswa sesuai dengan mata kuliah yang diprogram oleh mahasiswa dan dosen pengajar mata kuliah tersebut. Data type definition dari tabel d\_hadir ditunjukkan pada Tabel 4.9.

Tabel 4.9 Tabel data type definition dari tabel d\_hadir

Table Name	No	Field Name	Type	Key	Null	Width
d_hadir	01	id_hadir	smallint	PK	not null	
	02	id_jadwal	smallint	FK1	not null	
	03	nim	varchar	FK2	not null	20
	04	tanggal	date			
	05	jam	time			

#### 4.3.2.2 Data Field Description

Data field description menjelaskan keterangan seluruh kolom (field) dalam masing-masing tabel basis data kehadiran.

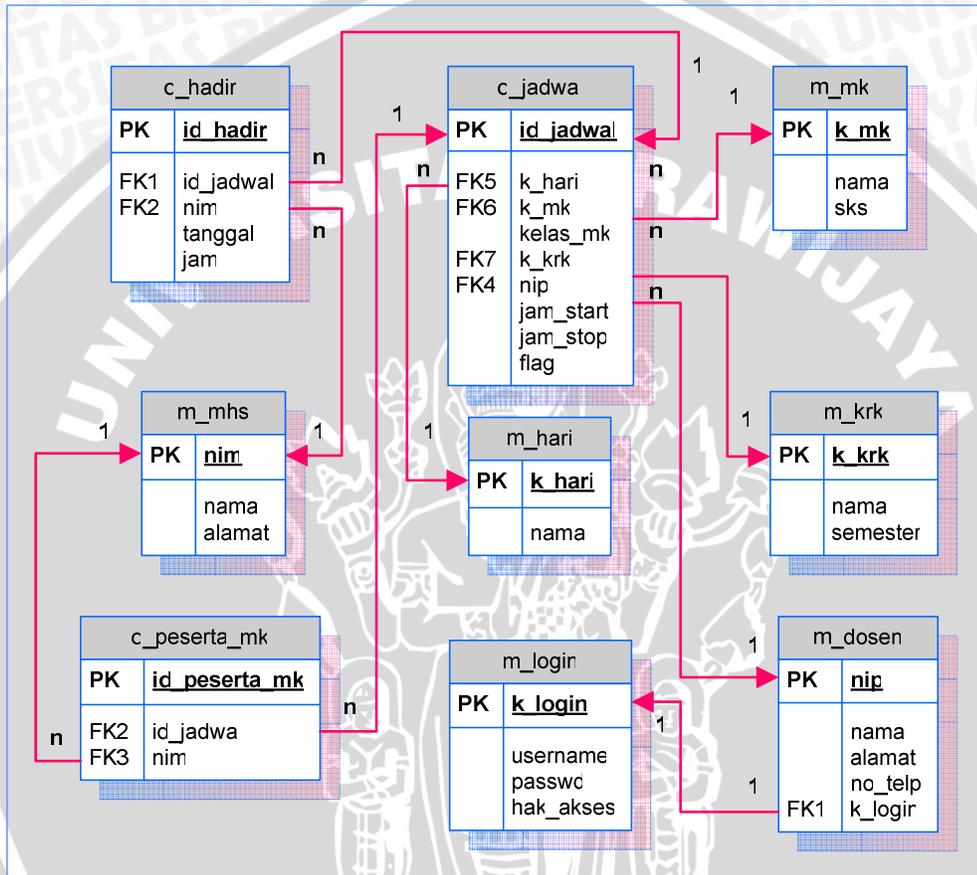
Data field description basis data kehadiran ditunjukkan pada Tabel 4.10.

**Tabel 4.10** Tabel data field description basis data kehadiran

Table Name	No	Field Name	Field Description
m_mhs	01	nim	nomor induk mahasiswa
	02	nama	nama lengkap mahasiswa
	03	alamat	alamat terakhir mahasiswa
m_dosen	01	nip	nomor induk pegawai (dosen)
	02	nama	nama lengkap dosen
	03	alamat	alamat rumah dosen
	04	no_telp	nomor telepon dosen
m_mk	01	k_mk	kode mata kuliah
	02	nama	nama mata kuliah
	03	sks	sks mata kuliah
m_krk	01	k_krk	kode kurikulum
	02	nama	tahun ajaran
	03	semester	semester kurikulum (genap/ganjil)
	04	krk_now	kurikulum yang berlaku saat ini
m_login	01	k_login	kode login
	02	username	username login
	03	passwd	password login
	04	hak_akses	hak akses login (admin/dosen)
m_hari	01	k_hari	kode hari
	02	nama	nama hari
d_peserta_mk	01	id_peserta_mk	id peserta mata kuliah
	02	nim	nomor induk mahasiswa
	03	id_jadwal	id jadwal mata kuliah
d_jadwal	01	id_jadwal	id jadwal mata kuliah
	02	nip	nomor induk pegawai (dosen)
	03	k_mk	kode mata kuliah
	04	k_krk	kode kurikulum
	05	kelas_mk	kelas mata kuliah
	06	k_hari	kode hari
	07	jam_start	jam mulai mata kuliah
	08	jam_stop	jam selesai mata kuliah
d_hadir	01	id_hadir	id hadir mahasiswa
	02	id_jadwal	id jadwal mata kuliah
	03	nim	nomor induk mahasiswa
	04	tanggal	tanggal pencatatan kehadiran
	05	jam	jam pencatatan kehadiran

### 4.3.3 Desain Basis Data kehadiran Ternormalisasi

Perancangan *entity relationship diagram* yang telah dipisahkan menurut entitas, atribut, dan relasi menghasilkan desain basis data dengan struktur tabel yang ternormalisasi yang disesuaikan dengan kebutuhan aplikasi. Desain basis data kehadiran ditunjukkan dalam Gambar 4.10.

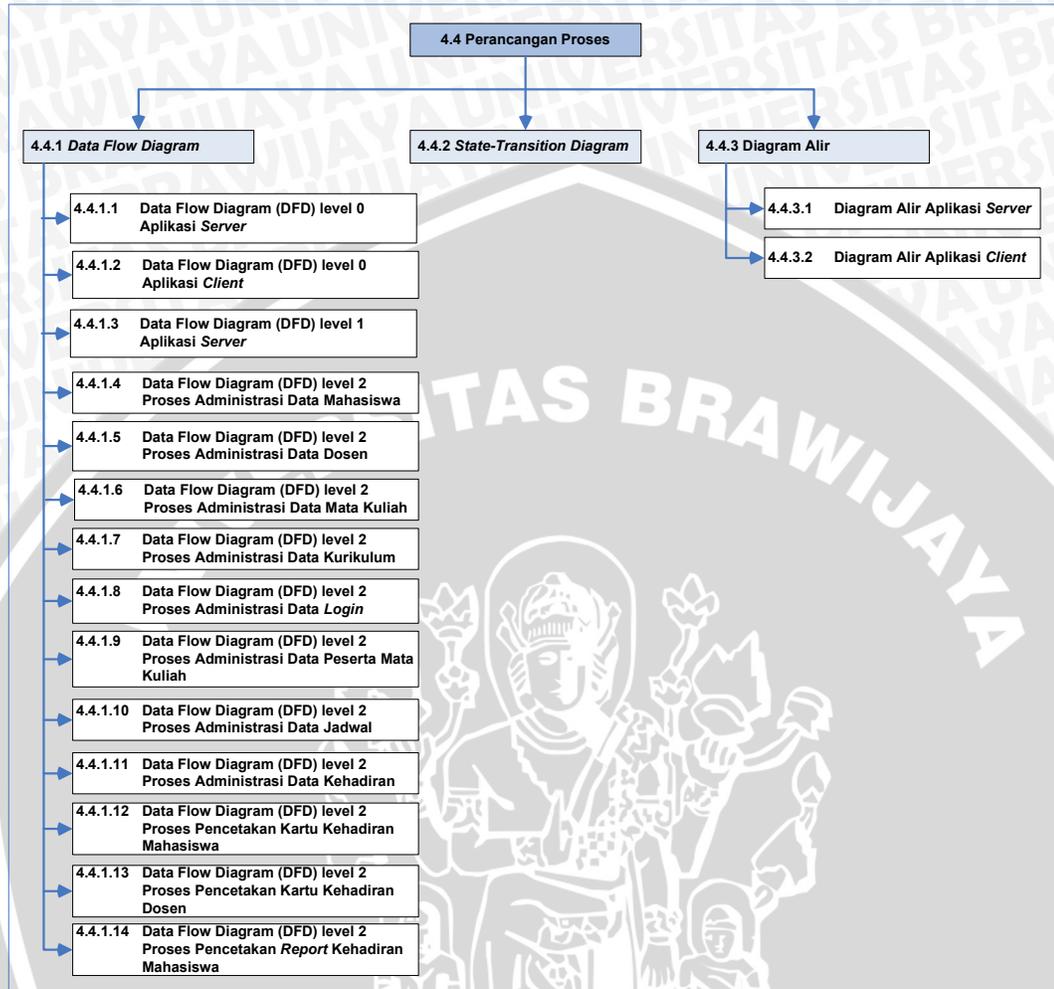


Gambar 4.10 Desain Basis Data kehadiran Ternormalisasi  
Sumber: Perancangan

### 4.4 Perancangan Proses

Perancangan proses menjelaskan masukan dan keluaran dari setiap proses yang terjadi pada Sistem Infomasi Kehadiran Mahasiswa. Perancangan proses yang dilakukan meliputi *Data Flow Diagram* (DFD), *State Transition Diagram* (STD) dan diagram alir proses.

Diagram pohon dari perancangan proses ditunjukkan dalam Gambar 4.11.



Gambar 4.11 Diagram Pohon Perancangan Proses  
Sumber: Perancangan

#### 4.4.1 Data Flow Diagram (DFD)

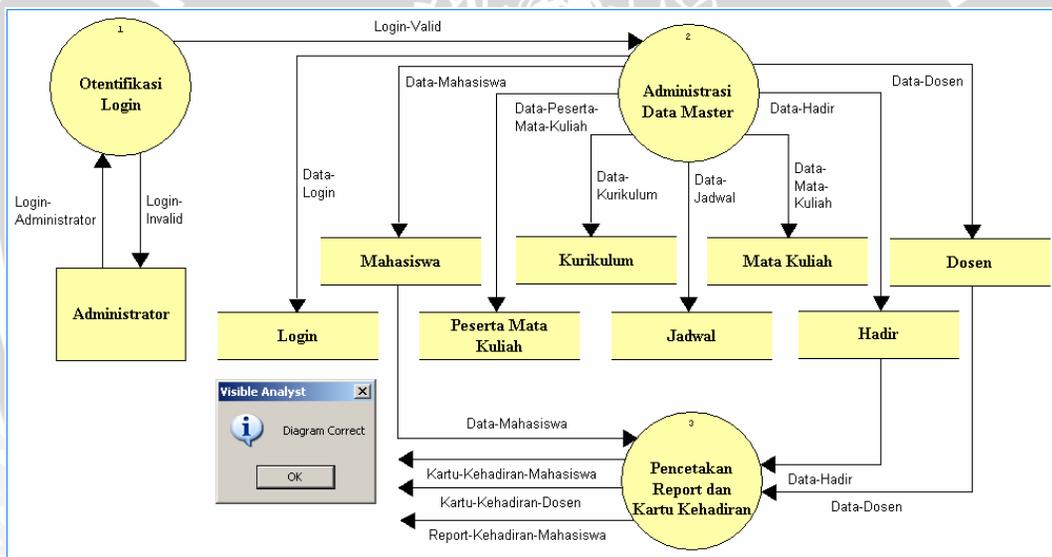
DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem [KRI-04:66]. DFD dirancang menggunakan *software* Visible Analyst 7.5 dengan aturan Yourdon / DeMarco.

Diagram konteks menggambarkan proses yang terjadi secara umum. Diagram konteks dari Sistem Informasi Kehadiran Mahasiswa ini ditunjukkan dalam Gambar 4.7. Proses di dalam diagram konteks dapat dijabarkan dengan menggunakan *data flow diagram*.

Sistem Informasi Kehadiran Mahasiswa (SIKMA) terdiri dari dua bagian utama, yaitu aplikasi *server* dan aplikasi *client*. Setiap aplikasi utama terdiri dari proses yang dijelaskan pada *data flow diagram level 0*. Pengguna dari aplikasi *server* adalah administrator yang mengatur seluruh administrasi data pada *server*. Pengguna dari aplikasi *client* adalah *user* yang terdiri dari dosen dan mahasiswa. Penjelasan mengenai pengguna aplikasi sesuai dengan yang dijelaskan dalam sub-bab 4.2.2 mengenai diagram konteks.

**4.4.1.1 Data Flow Diagram Level 0 Aplikasi Server**

DFD *level 0* merupakan penjabaran dari diagram konteks yang dapat dijabarkan lebih lanjut pada *level* berikutnya [TAT-04:166]. DFD *level 0* mencantumkan entitas yang berhubungan dengan sistem, ditambah dengan tabel penyimpanan data. DFD *level 0* aplikasi *server* ditunjukkan dalam Gambar 4.12.

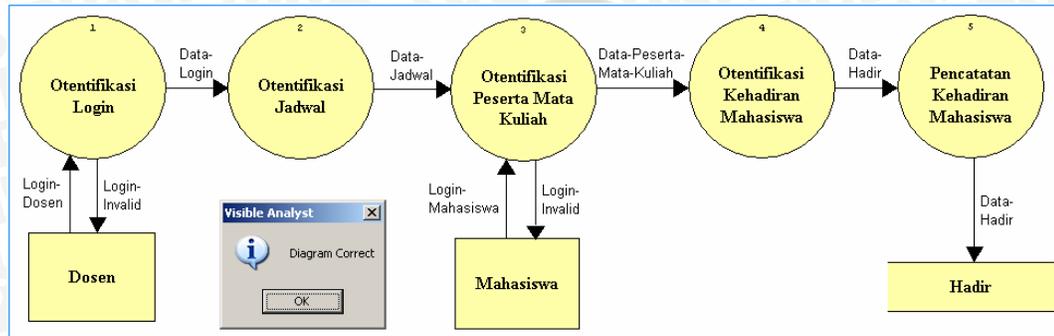


Gambar 4.12 Data Flow Diagram level 0 Aplikasi Server  
 Sumber: Perancangan

**4.4.1.2 Data Flow Diagram Level 0 Aplikasi Client**

Aplikasi *client* berfungsi untuk melakukan pencatatan kehadiran mahasiswa. Aplikasi *client* dipisah berdasarkan pengguna, yaitu dosen dan mahasiswa. Dosen melakukan autentikasi *login* untuk pengecekan NIP dosen sesuai yang terdapat pada tabel *m\_dosen*. Pengecekan jadwal mengajar mata kuliah dan kelas mata kuliah dilakukan setelah pengecekan NIP dosen pada proses *login*. Mahasiswa melakukan autentikasi peserta mata kuliah untuk mengecek apakah mahasiswa yang bersangkutan

telah terdaftar dalam mata kuliah sesuai dengan proses otentifikasi jadwal. Otentifikasi kehadiran mahasiswa berfungsi untuk memeriksa apakah mahasiswa telah tercatat kehadirannya. Mahasiswa dicatat kehadirannya setelah melalui pengecekan kehadiran mahasiswa. Proses yang terjadi pada aplikasi *client* ditunjukkan dalam Gambar 4.13.



**Gambar 4.13** Data Flow Diagram level 0 Aplikasi Client  
Sumber: Perancangan

Aplikasi *client* mempunyai lima macam proses, yaitu:

1. Proses 1 – Otentifikasi *Login*

Proses otentifikasi *login* merupakan proses pengecekan masukan Nomor Induk Pegawai (NIP) dosen dan hak akses dibandingkan dengan yang terdapat pada tabel *m\_dosen* dan *m\_login* di basis data kehadiran. Dosen melakukan proses *login* dengan Kartu Kehadiran Dosen yang akan dibaca oleh *barcode scanner*. Masukan NIP dosen dari *barcode scanner* akan dibandingkan dengan NIP dosen yang terdapat pada tabel *m\_dosen* di basis data kehadiran. Pesan kesalahan akan ditampilkan bila NIP dosen yang dimasukkan tidak sesuai dengan NIP dosen yang terdapat pada tabel *m\_dosen*. Pesan kesalahan juga ditampilkan apabila NIP dosen bernilai benar tetapi tidak mempunyai hak akses terhadap aplikasi *user*.

2. Proses 2 – Otentifikasi Jadwal Mata Kuliah

Proses otentifikasi jadwal merupakan proses pengecekan jadwal mata kuliah dan kelas mata kuliah sesuai dengan proses otentifikasi *login* berdasarkan hari dan jam dosen mengajar pada tabel *d\_jadwal*. Pesan kesalahan akan ditampilkan bila dosen tidak ada jadwal mengajar pada hari dan jam saat proses otentifikasi jadwal.

### 3. Proses 3 – Otentifikasi Peserta Mata Kuliah

Proses otentifikasi peserta mata kuliah merupakan proses pengecekan daftar peserta mata kuliah pada tabel `d_peserta_mk` sesuai dengan masukan dari nomor induk mahasiswa dalam bentuk *barcode*. Pesan kesalahan akan ditampilkan bila nomor induk mahasiswa tidak terdapat dalam tabel `d_peserta_mk` berdasarkan proses otentifikasi jadwal.

### 4. Proses 4 – Otentifikasi Kehadiran Mahasiswa

Proses otentifikasi kehadiran mahasiswa merupakan proses pengecekan pada tabel `d_hadir` sehingga mahasiswa tidak melakukan pencatatan kehadiran mahasiswa lebih dari satu kali pada mata kuliah yang sedang aktif.

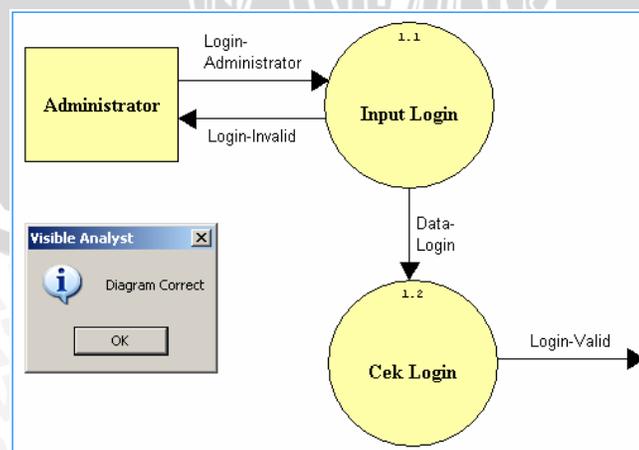
### 5. Proses 5 – Pencatatan Kehadiran Mahasiswa

Proses pencatatan kehadiran merupakan proses menyimpan data kehadiran mahasiswa pada tabel `d_hadir`. Informasi kehadiran mahasiswa akan ditampilkan pada aplikasi *client* setelah melalui proses otentifikasi peserta mata kuliah.

#### 4.4.1.3 Data Flow Diagram level 1 Aplikasi Server

Aplikasi *server* mempunyai tiga proses yaitu proses otentifikasi *login*, proses administrasi data master, dan proses pencetakan *report* dan kartu kehadiran.

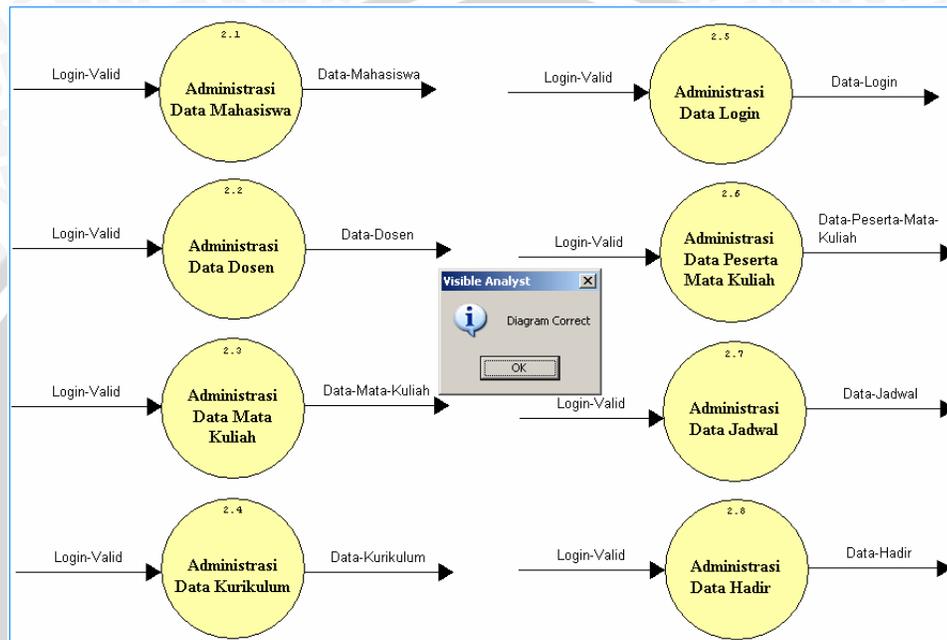
Proses otentifikasi *login* dibagi menjadi dua proses, yaitu *input login* dan cek *login*. *Input login* menerima masukan *login* dari administrator dan data masukan *login* akan dibandingkan dengan data *login* yang ada pada tabel `m_login`. *Data flow diagram level 1* aplikasi *server* untuk proses otentifikasi *login* ditunjukkan dalam Gambar 4.14.



Gambar 4.14 Data Flow Diagram level 1 Otentifikasi Login

Sumber: Perancangan

Proses administrasi data master terdiri dari administrasi data mahasiswa, dosen, mata kuliah, *login*, peserta mata kuliah, jadwal mata kuliah, kehadiran mahasiswa, pencetakan kartu kehadiran mahasiswa, pencetakan kartu kehadiran dosen dan pencetakan *report* kehadiran mahasiswa. *Data flow diagram level 1* aplikasi *server* untuk proses administrasi data master ditunjukkan dalam Gambar 4.15.



**Gambar 4.15** *Data Flow Diagram level 1* Administrasi Data Master  
Sumber: Perancangan

Administrasi data master terdiri dari delapan macam proses, yaitu:

1. Proses 2.1 – Administrasi Data Mahasiswa

Proses administrasi data mahasiswa merupakan proses manajemen data mahasiswa pada tabel *m\_mhs* di basis data kehadiran. Proses manajemen data mahasiswa dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.1.

2. Proses 2.2 – Administrasi Data Dosen

Proses administrasi data dosen merupakan proses manajemen data dosen pada tabel *m\_dosen* di basis data kehadiran. Proses manajemen data dosen dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.2.

3. Proses 2.3 – Administrasi Data Mata Kuliah

Proses administrasi data mata kuliah merupakan proses manajemen data mata kuliah pada tabel *m\_mk* di basis data kehadiran. Proses manajemen data mata kuliah dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.3.

#### 4. Proses 2.4 – Administrasi Data Kurikulum

Proses administrasi data kurikulum merupakan proses manajemen data kurikulum pada tabel *m\_krk* di basis data kehadiran. Proses manajemen data kurikulum dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.4.

#### 5. Proses 2.5 – Administrasi Data Login

Proses administrasi data *login* merupakan proses manajemen data *login* pada tabel *m\_login* di basis data kehadiran. Proses manajemen data *login* dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.5.

#### 6. Proses 2.6 – Administrasi Data Peserta Mata Kuliah

Proses administrasi data peserta mata kuliah merupakan proses manajemen data peserta mata kuliah pada tabel *d\_peserta\_mk* di basis data kehadiran. Proses manajemen data peserta mata kuliah dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.6.

#### 7. Proses 2.7 – Administrasi Data Jadwal Mata Kuliah

Proses administrasi data jadwal mata kuliah merupakan proses manajemen data jadwal mata kuliah pada tabel *d\_jadwal* di basis data kehadiran. Proses manajemen data jadwal mata kuliah dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.7.

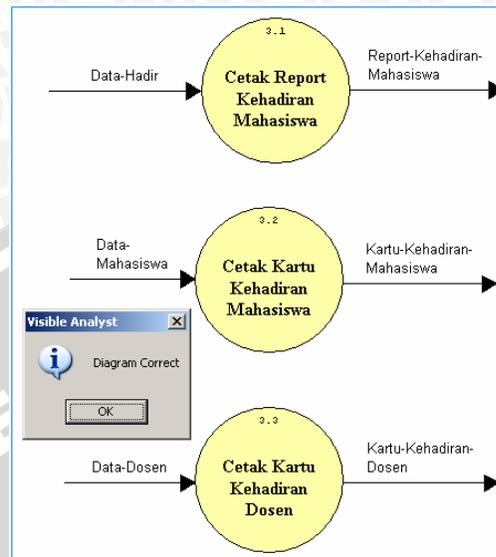
#### 8. Proses 2.8 – Administrasi Data Kehadiran Mahasiswa

Proses administrasi data kehadiran mahasiswa merupakan proses manajemen data kehadiran mahasiswa pada tabel *d\_jadwal* di basis data kehadiran. Proses manajemen data kehadiran mahasiswa dijelaskan pada *Data Flow Diagram level 2* terhadap proses 2.8.

Proses pencetakan *report* dan kartu kehadiran dibagi menjadi tiga, yaitu proses cetak *report* kehadiran mahasiswa, proses cetak kartu kehadiran mahasiswa, dan proses cetak kartu kehadiran dosen.

Proses pencetakan *report* kehadiran mahasiswa berfungsi untuk mencetak laporan kehadiran mahasiswa berdasarkan tabel *d\_hadir*. Proses pencetakan KKM berfungsi untuk mencetak kartu berisi nama mahasiswa, Nomor Induk Mahasiswa (NIM), dan NIM dalam bentuk *barcode* berdasarkan tabel *m\_mhs*. Proses pencetakan KKD berfungsi untuk mencetak kartu berisi Nomor Induk Pegawai (NIP), nama dosen, dan NIP dalam bentuk *barcode* berdasarkan tabel *m\_dosen*.

*Data flow diagram level 1* aplikasi *server* untuk proses pencetakan *report* dan kartu kehadiran ditunjukkan dalam Gambar 4.16.



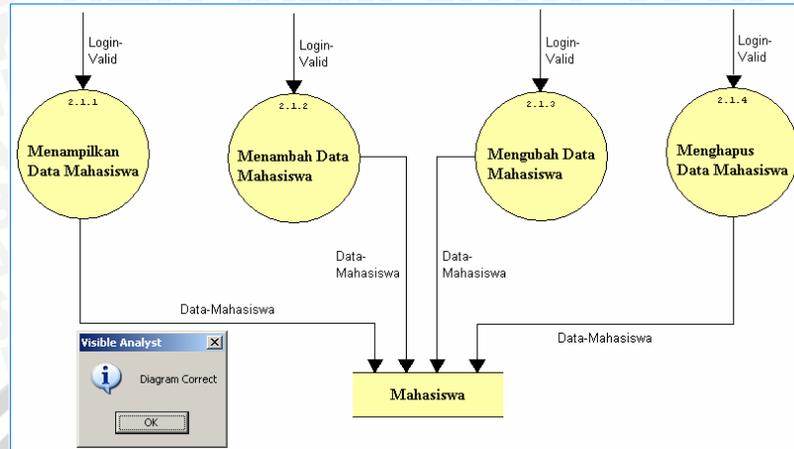
**Gambar 4.16** *Data Flow Diagram level 1* Pencetakan *Report* dan Kartu Kehadiran  
Sumber: Perancangan

1. Proses 3.1 – Cetak *Report* Kehadiran Mahasiswa  
Proses cetak *report* kehadiran mahasiswa merupakan proses mencetak *report* kehadiran mahasiswa berdasarkan tabel `d_hadir`.
2. Proses 3.1 – Cetak Kartu Kehadiran Mahasiswa  
Proses cetak Kartu Kehadiran Mahasiswa merupakan proses mencetak Kartu Kehadiran Mahasiswa berdasarkan tabel `m_mhs`.
3. Proses 3.3 – Cetak Kartu Kehadiran Dosen  
Proses cetak Kartu Kehadiran Dosen merupakan proses mencetak Kartu Kehadiran Dosen berdasarkan tabel `m_dosen`.

#### 4.4.1.4 *Data Flow Diagram level 2* Proses Administrasi Data Mahasiswa

Proses 2.1 dalam Gambar 4.15 merupakan proses administrasi data mahasiswa sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data mahasiswa berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data mahasiswa pada tabel `m_mhs`.

Proses administrasi data mahasiswa ditunjukkan dalam Gambar 4.17.



**Gambar 4.17** DFD level 2 Proses 2.1 Administrasi Data Mahasiswa  
Sumber: Perancangan

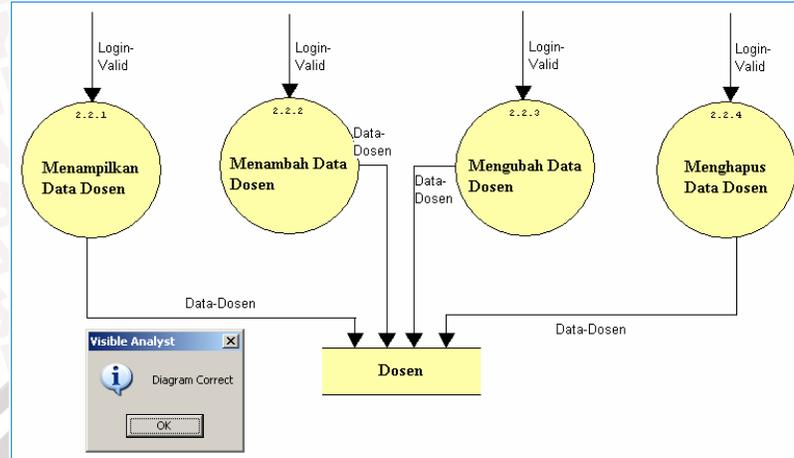
Proses administrasi data mahasiswa mempunyai empat proses, yaitu:

1. Proses 2.1.1 – Menampilkan Data Mahasiswa  
Proses menampilkan data mahasiswa adalah proses melihat data mahasiswa pada tabel `m_mhs`.
2. Proses 2.1.2 – Menambah Data Mahasiswa  
Proses menambah data mahasiswa adalah proses menambah data mahasiswa pada tabel `m_mhs`.
3. Proses 2.1.3 – Mengubah Data Mahasiswa  
Proses mengubah data mahasiswa adalah proses mengubah data mahasiswa pada tabel `m_mhs`.
4. Proses 2.1.4 – Menghapus Data Mahasiswa  
Proses menghapus data mahasiswa adalah proses menghapus data mahasiswa pada tabel `m_mhs`.

#### 4.4.1.5 Data Flow Diagram level 2 Proses Administrasi Data Dosen

Proses 2.2 dalam Gambar 4.15 merupakan proses administrasi data dosen sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data dosen berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data dosen pada tabel `m_dosen`.

Proses administrasi data dosen ditunjukkan dalam Gambar 4.18.



**Gambar 4.18** DFD level 2 Proses 2.2 Administrasi Data Dosen  
Sumber: Perancangan

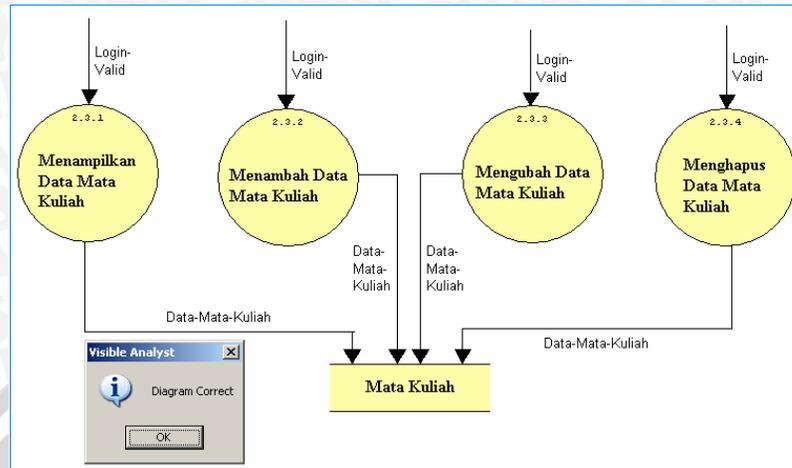
Proses administrasi data dosen mempunyai empat proses, yaitu:

1. Proses 2.2.1 – Menampilkan Data Dosen  
Proses menampilkan data dosen adalah proses melihat data dosen pada tabel `m_dosen`.
2. Proses 2.2.2 – Menambah Data Dosen  
Proses menambah data dosen adalah proses menambah data dosen pada tabel `m_dosen`.
3. Proses 2.2.3 – Mengubah Data Dosen  
Proses mengubah data dosen adalah proses mengubah data dosen pada tabel `m_dosen`.
4. Proses 2.2.4 – Menghapus Data Dosen  
Proses menghapus data dosen adalah proses menghapus data dosen pada tabel `m_dosen`.

#### 4.4.1.6 Data Flow Diagram level 2 Proses Administrasi Data Mata Kuliah

Proses 2.3 dalam Gambar 4.15 merupakan proses administrasi mata kuliah sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data mata kuliah berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data mata kuliah pada tabel `m_mk`.

Proses administrasi data mata kuliah ditunjukkan dalam Gambar 4.19.



**Gambar 4.19** DFD level 2 Proses 2.3 Administrasi Data Mata Kuliah  
Sumber: Perancangan

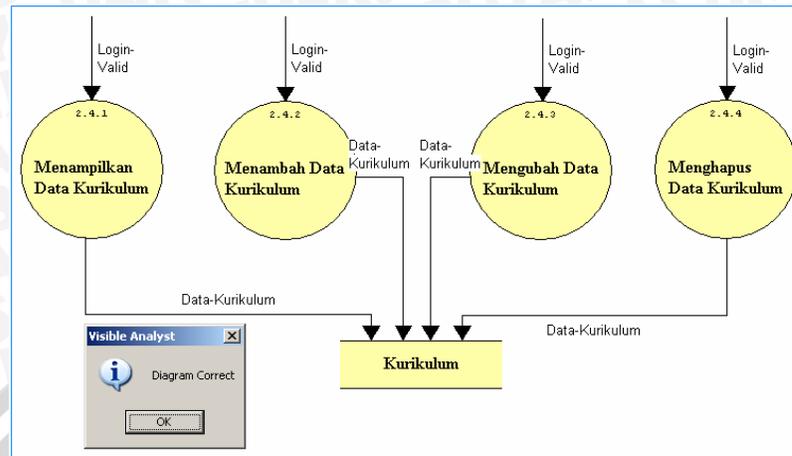
Proses administrasi data mata kuliah mempunyai empat proses, yaitu:

1. Proses 2.3.1 – Menampilkan Data Mata Kuliah  
Proses menampilkan data mata kuliah adalah proses melihat data mata kuliah pada tabel *m\_mk*.
2. Proses 2.3.2 – Menambah Data Mata Kuliah  
Proses menambah data mata kuliah adalah proses menambah data mata kuliah pada tabel *m\_mk*.
3. Proses 2.3.3 – Mengubah Data Mata Kuliah  
Proses mengubah data mata kuliah adalah proses mengubah data mata kuliah pada tabel *m\_mk*.
4. Proses 2.3.4 – Menghapus Data Mata Kuliah  
Proses menghapus data mata kuliah adalah proses menghapus data mata kuliah pada tabel *m\_mk*.

#### 4.4.1.7 Data Flow Diagram level 2 Proses Administrasi Data Kurikulum

Proses 2.4 dalam Gambar 4.15 merupakan proses administrasi data kurikulum sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data kurikulum berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data kurikulum pada tabel *m\_krk*.

Proses administrasi data kurikulum ditunjukkan dalam Gambar 4.20.



**Gambar 4.20** DFD level 2 Proses 2.4 Administrasi Kurikulum  
Sumber: Perancangan

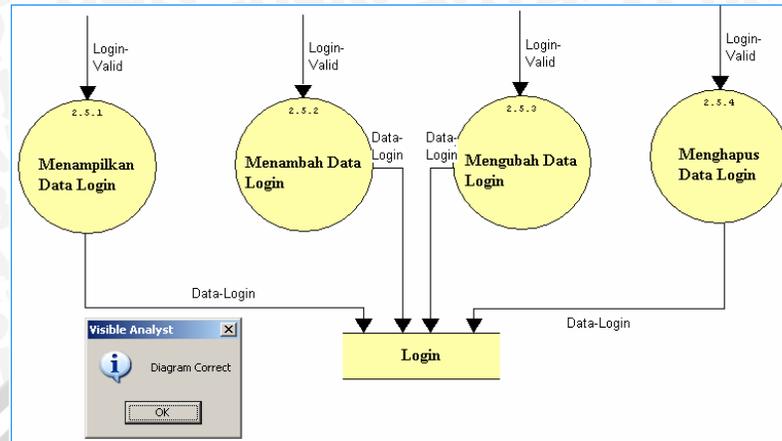
Proses administrasi data kurikulum mempunyai empat proses, yaitu:

1. Proses 2.4.1 – Menampilkan Data Kurikulum  
Proses menampilkan data kurikulum adalah proses melihat data kurikulum pada tabel `m_krk`.
2. Proses 2.4.2 – Menambah Data Kurikulum  
Proses menambah data kurikulum adalah proses menambah data kurikulum pada tabel `m_krk`.
3. Proses 2.4.3 – Mengubah Data Kurikulum  
Proses mengubah data kurikulum adalah proses mengubah data kurikulum pada tabel `m_krk`.
4. Proses 2.4.4 – Menghapus Data Kurikulum  
Proses menghapus data kurikulum adalah proses menghapus data kurikulum pada tabel `m_krk`.

#### 4.4.1.8 Data Flow Diagram level 2 Proses Administrasi Data Login

Proses 2.5 dalam Gambar 4.15 merupakan proses administrasi data mahasiswa sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data *login* berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data *login* pada tabel `m_login`. Proses administrasi data *login* menampilkan data *login* dan data dosen yang mempunyai relasi dengan tabel `m_dosen`.

Proses administrasi data *login* ditunjukkan dalam Gambar 4.21.



**Gambar 4.21** DFD level 2 Proses 2.5 Administrasi Data *Login*  
Sumber: Perancangan

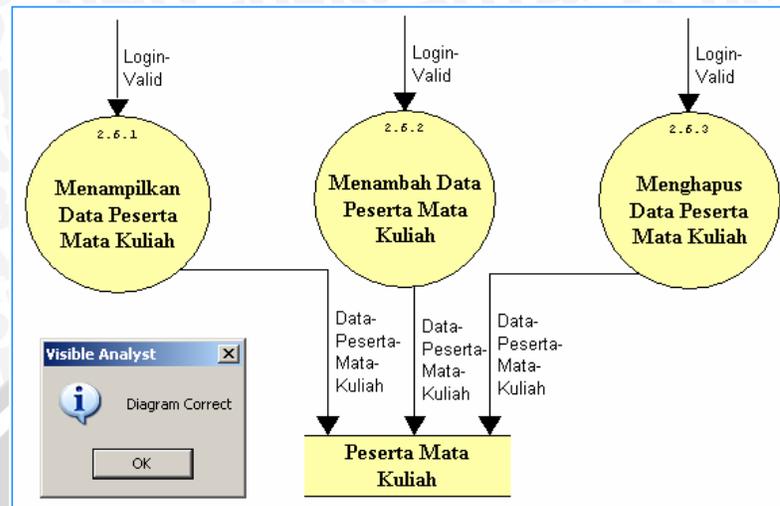
Proses administrasi data *login* mempunyai empat proses, yaitu:

1. Proses 2.5.1 – Menampilkan Data *Login*  
Proses menampilkan data *login* adalah proses melihat data *login* pada tabel *m\_login*.
2. Proses 2.5.2 – Menambah Data *Login*  
Proses menambah data *login* adalah proses menambah data *login* pada tabel *m\_login*.
3. Proses 2.5.3 – Mengubah Data *Login*  
Proses mengubah data *login* adalah proses mengubah data *login* pada tabel *m\_login*.
4. Proses 2.5.4 – Menghapus Data *Login*  
Proses menghapus data *login* adalah proses menghapus data *login* pada tabel *m\_login*.

#### 4.4.1.9 Data Flow Diagram level 2 Proses Administrasi Data Peserta Mata Kuliah

Proses 2.6 dalam Gambar 4.15 merupakan proses administrasi data peserta mata kuliah sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data peserta mata kuliah berfungsi untuk menambah, menghapus, mengubah data pada tabel *d\_peserta\_mk*. Proses administrasi data peserta mata kuliah menampilkan data peserta mata kuliah dari tabel *d\_peserta\_mk*.

Proses administrasi data peserta mata kuliah ditunjukkan dalam Gambar 4.22.



**Gambar 4.22** DFD level 2 Proses 2.6 Administrasi Data Peserta Mata Kuliah  
Sumber: Perancangan

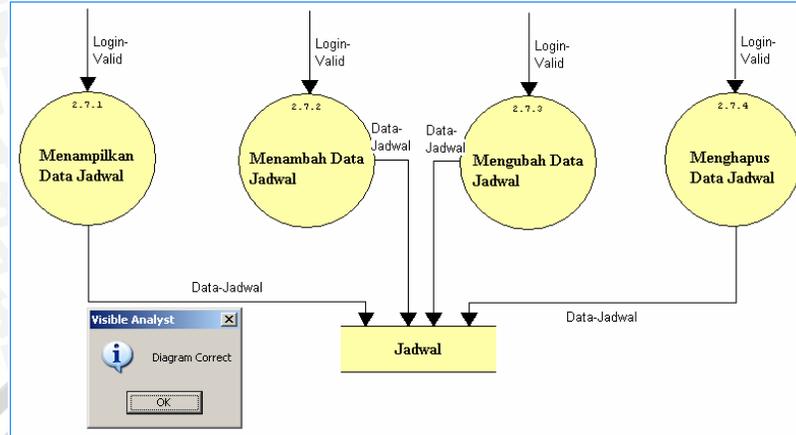
Proses administrasi data peserta mata kuliah mempunyai empat proses, yaitu:

1. Proses 2.6.1 – Menampilkan Data Peserta Mata Kuliah  
Proses menampilkan data peserta mata kuliah adalah proses melihat data peserta mata kuliah pada tabel `d_peserta_mk`.
2. Proses 2.6.2 – Menambah Data Peserta Mata Kuliah  
Proses menambah data peserta mata kuliah adalah proses menambah data peserta mata kuliah pada tabel `d_peserta_mk`.
3. Proses 2.6.4 – Menghapus Data Peserta Mata Kuliah  
Proses menghapus data peserta mata kuliah adalah proses menghapus data peserta mata kuliah pada tabel `d_peserta_mk`.

#### 4.4.1.10 Data Flow Diagram level 2 Proses Administrasi Data Jadwal Mata Kuliah

Proses 2.7 dalam Gambar 4.15 merupakan proses administrasi data jadwal mata kuliah sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data jadwal mata kuliah berfungsi untuk menambah, menghapus, mengubah data jadwal mata kuliah pada tabel `d_jadwal`. Proses administrasi data jadwal mata kuliah menampilkan data jadwal mata kuliah dari tabel `d_jadwal`.

Proses administrasi data jadwal mata kuliah ditunjukkan dalam Gambar 4.23.



**Gambar 4.23** DFD level 2 Proses 2.7 Administrasi Data Jadwal Mata Kuliah  
Sumber: Perancangan

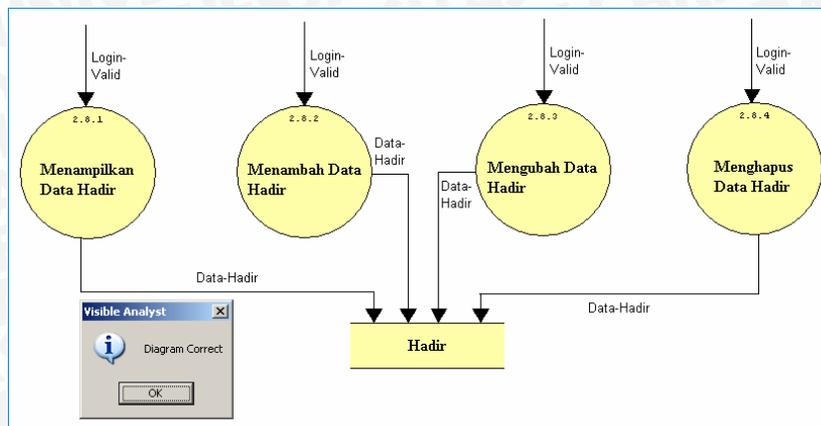
Proses administrasi data jadwal mata kuliah mempunyai empat proses, yaitu:

1. Proses 2.7.1 – Menampilkan Data Jadwal Mata Kuliah  
Proses menampilkan data jadwal mata kuliah adalah proses melihat data jadwal mata kuliah pada tabel `d_jadwal`.
2. Proses 2.7.2 – Menambah Data Jadwal Mata Kuliah  
Proses menambah data jadwal mata kuliah adalah proses menambah data jadwal mata kuliah pada tabel `d_jadwal`.
3. Proses 2.7.3 – Mengubah Data Jadwal Mata Kuliah  
Proses mengubah data jadwal mata kuliah adalah proses mengubah data jadwal mata kuliah pada tabel `d_jadwal`.
4. Proses 2.7.4 – Menghapus Data Jadwal Mata Kuliah  
Proses menghapus data jadwal mata kuliah adalah proses menghapus data jadwal mata kuliah pada tabel `d_jadwal`.

#### 4.4.1.11 Data Flow Diagram level 2 Proses Administrasi Data Kehadiran Mahasiswa

Proses 2.8 dalam Gambar 4.15 merupakan proses administrasi data kehadiran mahasiswa sesuai dengan *data flow diagram level 1* aplikasi data master. Proses administrasi data kehadiran mahasiswa berfungsi untuk menambah, menghapus, mengubah data kehadiran mahasiswa pada tabel `d_hadir`. Proses administrasi data kehadiran mahasiswa menampilkan data kehadiran mahasiswa dari tabel `d_hadir`.

Proses administrasi data kehadiran mahasiswa ditunjukkan dalam Gambar 4.24.



**Gambar 4.24** DFD level 2 Proses 2.8 Administrasi Data Kehadiran Mahasiswa  
Sumber: Perancangan

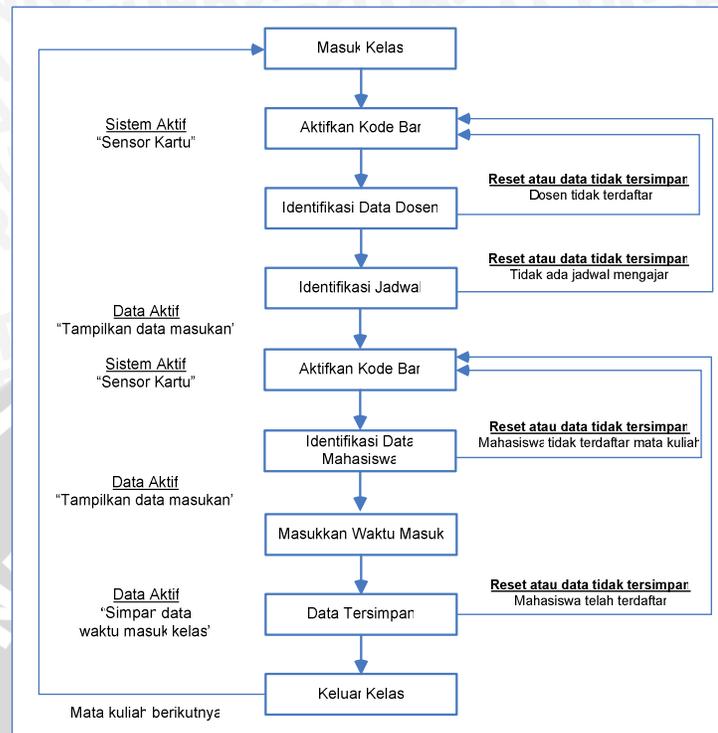
Proses administrasi data kehadiran mahasiswa mempunyai empat proses, yaitu:

1. Proses 2.8.1 – Menampilkan Data Kehadiran Mahasiswa  
Proses menampilkan data kehadiran mahasiswa adalah proses melihat data kehadiran mahasiswa pada tabel `d_hadir`.
2. Proses 2.8.2 – Menambah Data Kehadiran Mahasiswa  
Proses menambah data kehadiran mahasiswa adalah proses menambah data kehadiran mahasiswa pada tabel `d_hadir`.
3. Proses 2.8.3 – Mengubah Data Kehadiran Mahasiswa  
Proses mengubah data kehadiran mahasiswa adalah proses mengubah data kehadiran mahasiswa pada tabel `d_hadir`.
4. Proses 2.8.4 – Menghapus Data Kehadiran Mahasiswa  
Proses menghapus data kehadiran mahasiswa adalah proses menghapus data kehadiran mahasiswa pada tabel `d_hadir`.

#### 4.4.2 State Transition Diagram (STD) Proses Pencatatan Kehadiran Mahasiswa

*State Transition Diagram* (STD) berfungsi untuk memodelkan ketergantungan suatu sistem terhadap waktu. Sistem Informasi Kehadiran Mahasiswa mempunyai *state transition diagram* pada aplikasi *client* pada proses pencatatan kehadiran mahasiswa. Pencatatan kehadiran mahasiswa berdasarkan tanggal dan jam sesuai dengan tanggal dan jam pada aplikasi *server* merupakan proses *real-time system* untuk menjamin keakuratan data. Data kehadiran mahasiswa dicatat dan disimpan dalam basis data kehadiran secara langsung.

STD proses pencatatan kehadiran mahasiswa ditunjukkan dalam Gambar 4.25.



**Gambar 4.25** State Transition Diagram Proses Pencatatan Kehadiran Mahasiswa  
Sumber: Perancangan

Proses pencatatan kehadiran mahasiswa secara *real-time* dimulai pada saat proses otentifikasi *login* dosen. Dosen pengajar melakukan proses *login* dengan mengaktifkan *barcode* melalui kartu kehadiran dosen. *Login* dosen pengajar menentukan mata kuliah berdasarkan hari dan jam mengajar dosen sesuai dengan jadwal mata kuliah. Mahasiswa melakukan proses pencatatan kehadiran dengan mengaktifkan *barcode* melalui kartu kehadiran mahasiswa yang telah dicetak sebelumnya. Mahasiswa yang telah terdaftar pada kelas mata kuliah dapat melakukan proses pencatatan kehadiran mahasiswa. Data waktu kehadiran mahasiswa akan disimpan dalam basis data kehadiran. Proses pencatatan kehadiran akan diulang kembali ketika dosen memulai kelas mata kuliah baru.

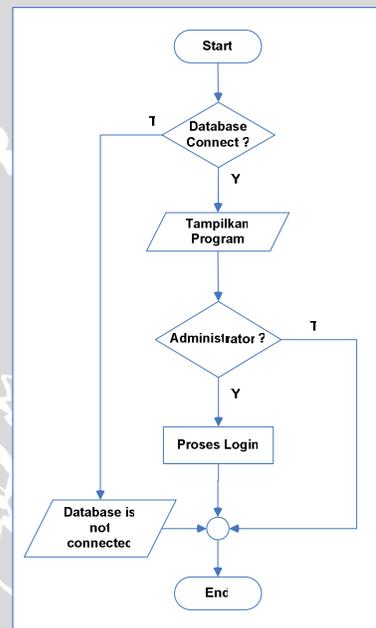
#### 4.4.3 Diagram Alir Sistem Informasi Mahasiswa

Diagram alir berfungsi untuk memodelkan masukan, keluaran, proses, dan transaksi dengan menggunakan simbol-simbol tertentu [KRI-04:82]. Diagram alir merupakan salah satu metode dari spesifikasi proses. Spesifikasi proses adalah pendeskripsian proses yang terjadi pada level paling dasar dalam DFD [KRI-04:79].

Diagram alir Sistem Informasi Kehadiran Mahasiswa dibagi menjadi dua bagian utama, yaitu diagram alir aplikasi *server* dan aplikasi *client*.

### 3.4.3.1 Diagram Alir Aplikasi *Server*

Diagram alir aplikasi *server* menjelaskan proses yang terjadi pada aplikasi *server*. Proses yang terjadi pada aplikasi *server* sesuai dengan yang dijelaskan pada sub-bab 4.4.1.1. Diagram alir aplikasi *server* ditunjukkan dalam Gambar 4.26.



Gambar 4.26 Diagram Alir Aplikasi *Server*

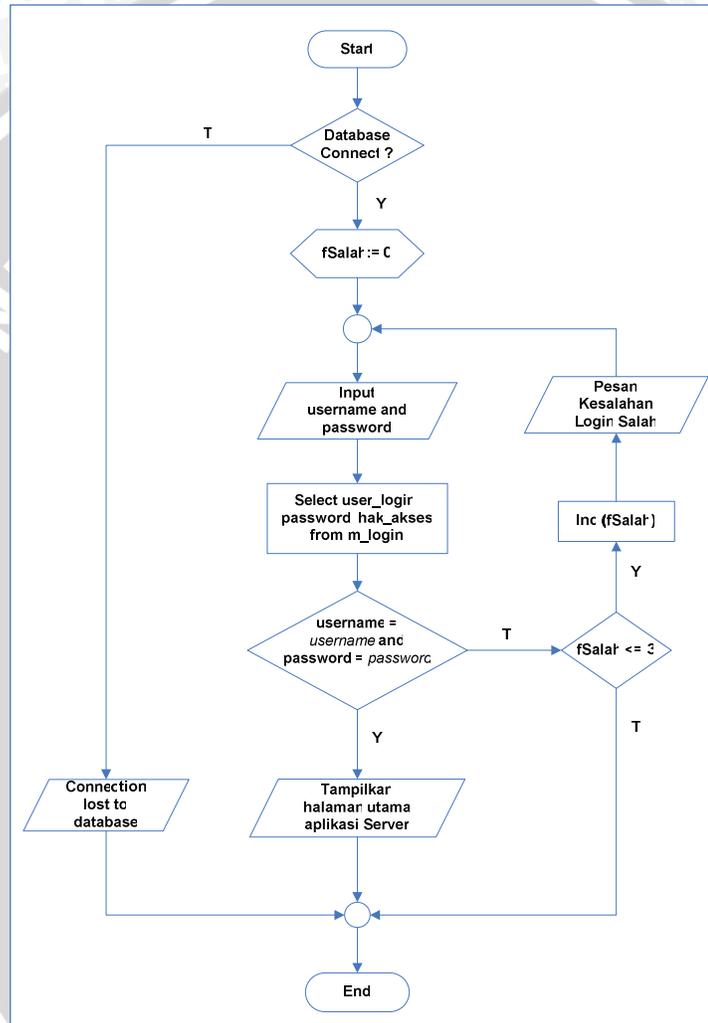
Sumber: *Perancangan*

Proses pertama yang terjadi setelah menjalankan `server.exe` adalah pengecekan koneksi pada basis data. Aplikasi *server* menampilkan pesan kesalahan bila *server* basis data tidak berjalan. Administrator adalah pengguna yang mempunyai hak akses untuk aplikasi *server*. Administrator melakukan proses *login* untuk menjalankan aplikasi *server*.

#### 4.4.3.1.1 Diagram Alir Proses *Login Server*

Proses pertama adalah memasukkan nilai ke dalam variabel `fSalah` sebagai penghitung kesalahan dalam proses *login*. Proses kedua adalah memasukkan *username* dan *password* untuk dibandingkan dengan *field username* dan *password* di tabel `m_login`. *Username* dan *password* yang sesuai pada tabel `m_login` akan menampilkan halaman utama aplikasi *server*. Proses menu aplikasi dijelaskan mulai sub-bab 4.4.1.4

sampai dengan sub-bab 4.4.1.14. Pesan kesalahan akan ditampilkan bila tidak ada kesamaan antara data masukan dengan data pada tabel *m\_login*. Aplikasi *server* akan tertutup bila pengguna melakukan kesalahan memasukkan *login* sebanyak tiga kali. Proses *login* dapat dilakukan kembali dengan membuka aplikasi *server* setelah melakukan kesalahan *login* sebanyak tiga kali. Diagram alir *login server* ditunjukkan dalam Gambar 4.27.

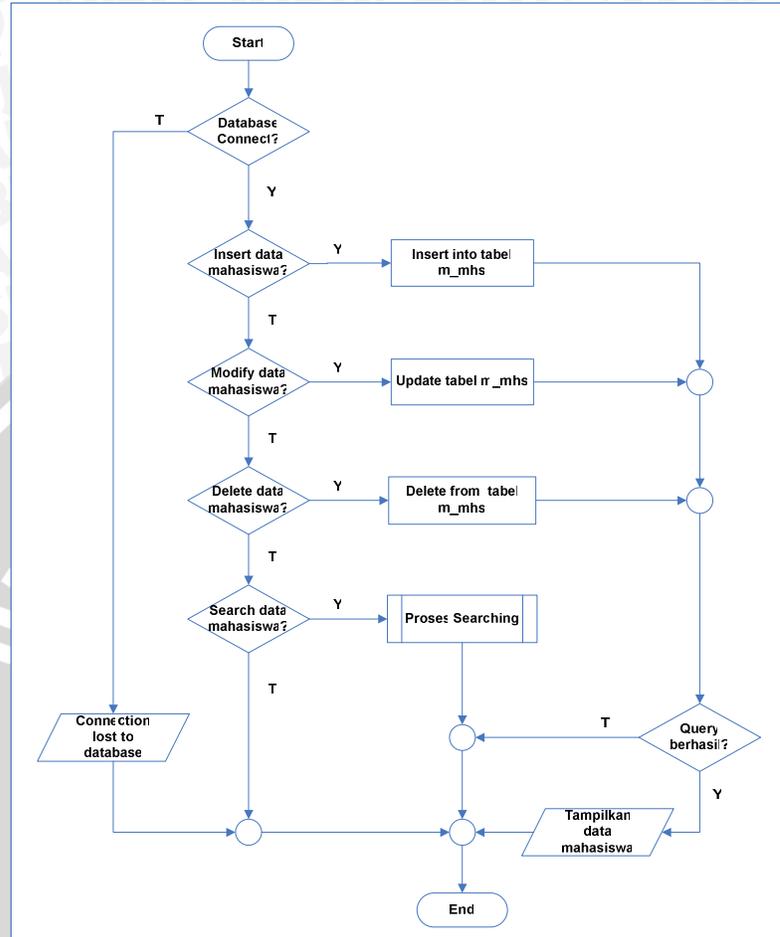


Gambar 4.27 Diagram Alir Login Server  
Sumber: Perancangan

#### 4.4.3.1.2 Diagram Alir Proses Administrasi Data Mahasiswa

Proses administrasi data mahasiswa mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data mahasiswa pada tabel *m\_mhs*.

Diagram alir administrasi data mahasiswa ditunjukkan dalam Gambar 4.28.



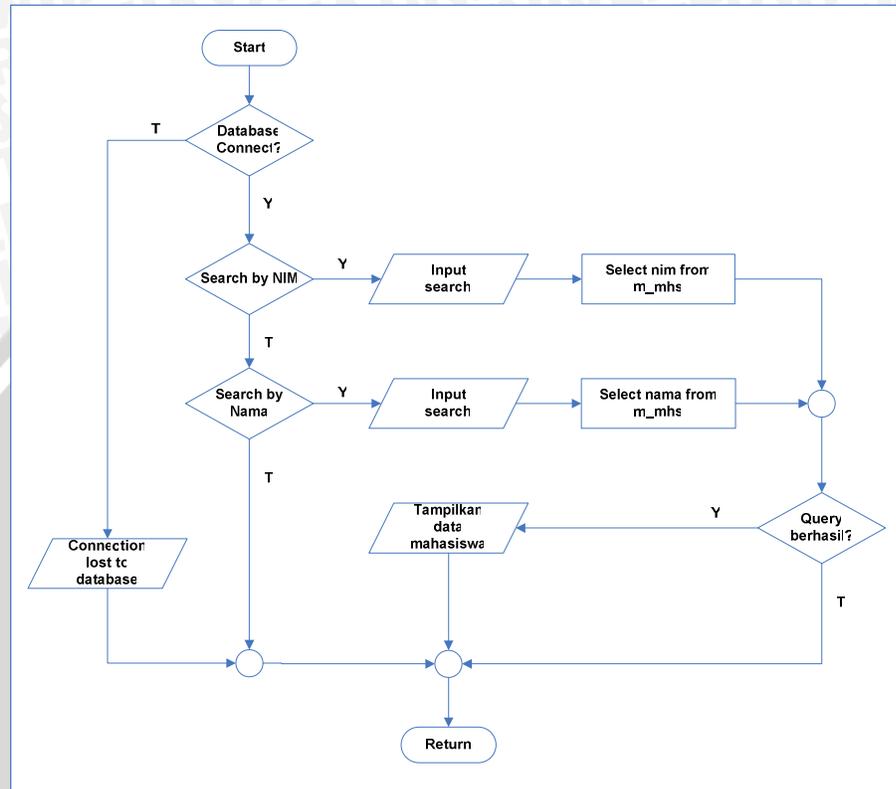
**Gambar 4.28** Diagram Alir Administrasi Data Mahasiswa  
Sumber: Perancangan

Proses yang terjadi pada administrasi data mahasiswa dijelaskan sebagai berikut:

1. Proses menambah data mahasiswa dilakukan dengan memasukkan data mahasiswa baru ke dalam *form* pendataan mahasiswa. Data mahasiswa yang dimasukkan akan disimpan dalam tabel *m\_mhs*.
2. Proses mengubah data mahasiswa dilakukan dengan mengubah data mahasiswa yang terdapat dalam *form* pendataan mahasiswa. Data mahasiswa yang diubah akan disimpan dalam tabel *m\_mhs*.
3. Proses menghapus data mahasiswa dilakukan dengan menghapus data mahasiswa terdapat dalam *form* pendataan mahasiswa. Data mahasiswa yang dihapus akan dihilangkan dari tabel *m\_mhs*.

4. Proses pencarian data mahasiswa merupakan proses untuk mencari data mahasiswa sesuai dengan kehendak administrator.

Diagram alir pencarian data mahasiswa ditunjukkan dalam Gambar 4.29.



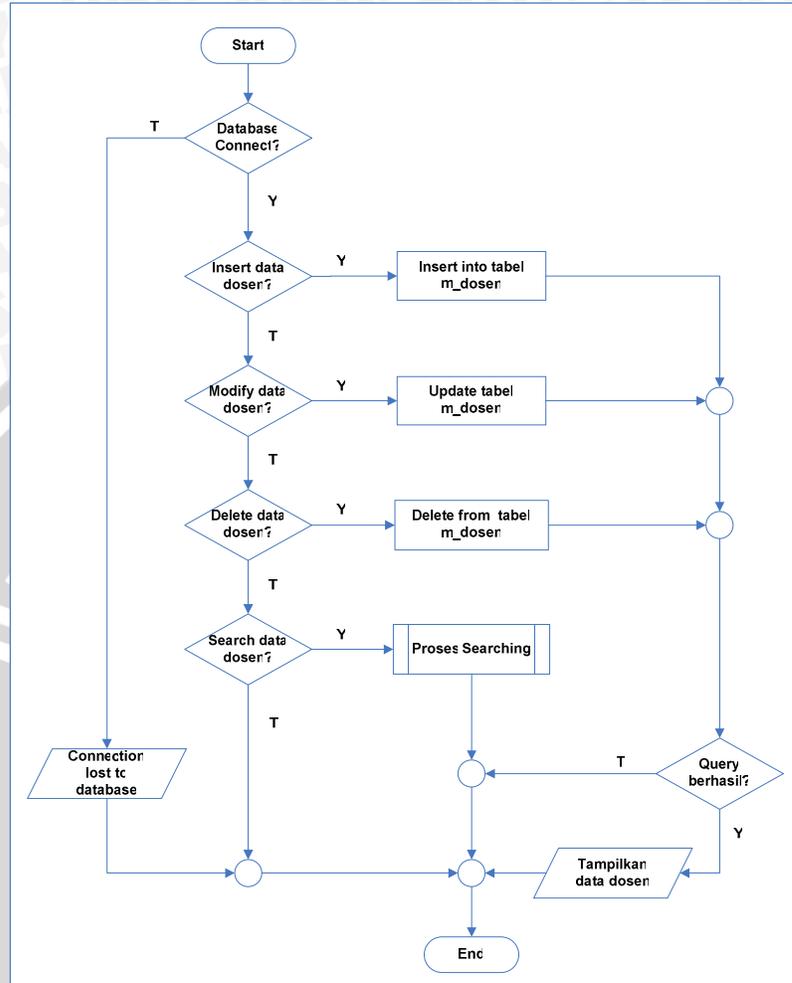
**Gambar 4.29** Diagram Alir Pencarian Data Mahasiswa  
Sumber: Perancangan

Proses pencarian data mahasiswa dipisahkan berdasarkan NIM mahasiswa dan nama mahasiswa. Pencarian NIM mahasiswa dilakukan dengan membandingkan masukan NIM mahasiswa pada *form* mahasiswa dengan NIM mahasiswa pada tabel *m\_mhs*. Pencarian nama mahasiswa dilakukan dengan membandingkan masukan nama mahasiswa pada *form* pendataan mahasiswa dengan nama mahasiswa pada tabel *m\_mhs*. Hasil pencarian data mahasiswa akan ditampilkan pada *form* pendataan mahasiswa.

#### 4.4.3.1.3 Diagram Alir Proses Administrasi Data Dosen

Proses administrasi data dosen mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data dosen pada tabel *m\_dosen*.

Diagram alir administrasi data dosen ditunjukkan dalam Gambar 4.30.



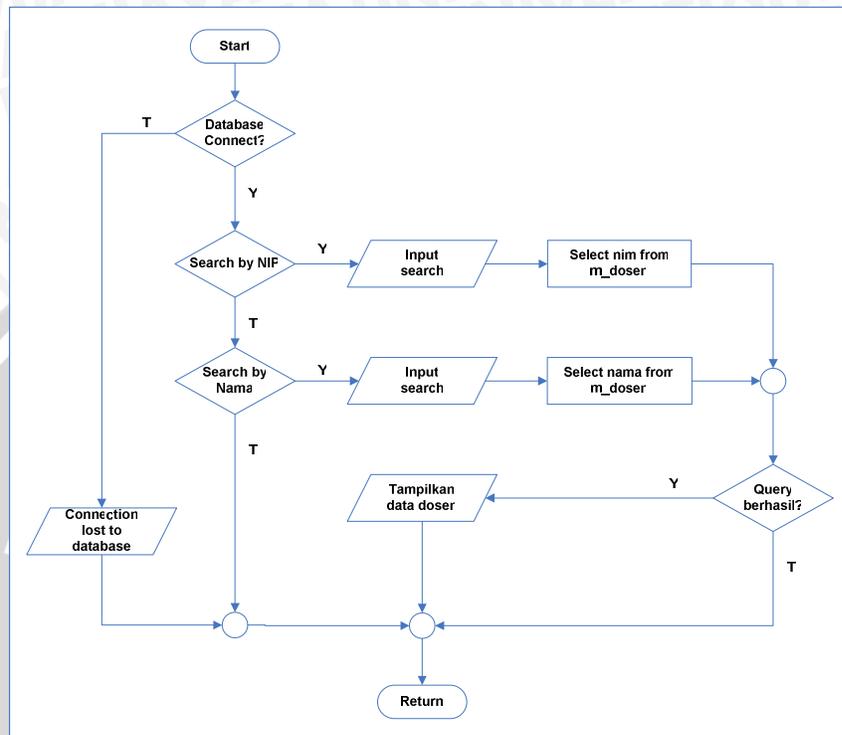
Gambar 4.30 Diagram Alir Administrasi Data Dosen  
Sumber: Perancangan

Proses yang terjadi pada administrasi data dosen dijelaskan sebagai berikut:

1. Proses menambah data dosen dilakukan dengan memasukkan data dosen baru ke dalam *form* pendataan dosen. Data dosen yang dimasukkan akan disimpan dalam tabel *m\_dosen*.
2. Proses mengubah data dosen dilakukan dengan mengubah data dosen yang terdapat dalam *form* pendataan dosen. Data dosen yang diubah akan disimpan dalam tabel *m\_dosen*.
3. Proses menghapus data dosen dilakukan dengan menghapus data dosen terdapat dalam *form* pendataan dosen. Data dosen yang dihapus akan dihilangkan dari tabel *m\_dosen*.

4. Proses pencarian data dosen merupakan proses untuk mencari data dosen sesuai dengan kehendak administrator.

Diagram alir pencarian data dosen ditunjukkan dalam Gambar 4.31.



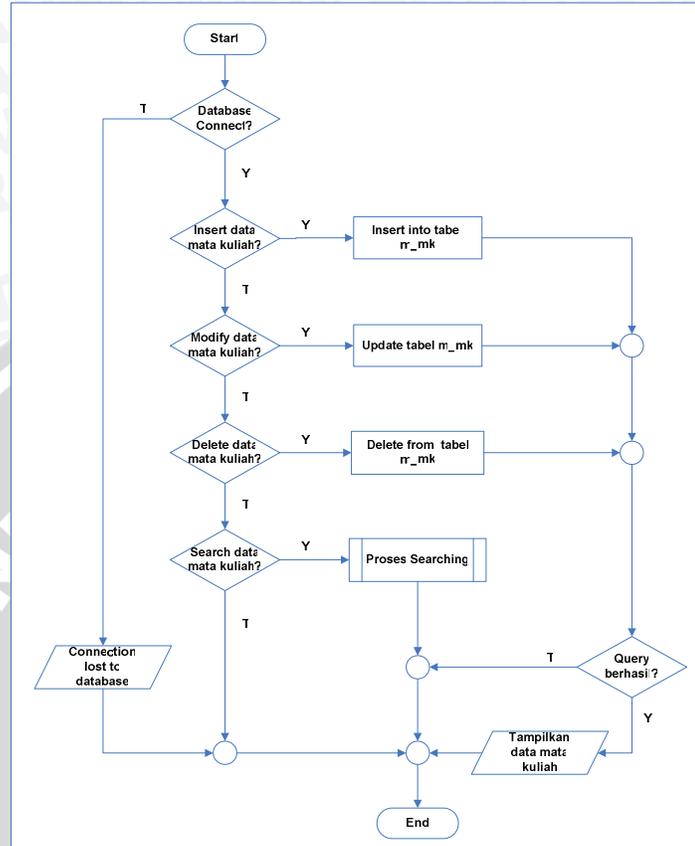
Gambar 4.31 Diagram Alir Pencarian Data Dosen  
Sumber: Perancangan

Proses pencarian data dosen dipisahkan berdasarkan NIP dosen dan nama dosen. Pencarian NIP dosen dilakukan dengan membandingkan masukan NIP dosen pada *form* dosen dengan NIP dosen pada tabel *m\_dosen*. Pencarian nama dosen dilakukan dengan membandingkan masukan nama dosen pada *form* pendataan dosen dengan nama dosen pada tabel *m\_dosen*. Hasil pencarian data dosen akan ditampilkan pada *form* pendataan dosen.

#### 4.4.3.1.4 Diagram Alir Proses Administrasi Data Mata Kuliah

Proses administrasi data mata kuliah mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data mata kuliah pada tabel *m\_mk*.

Diagram alir administrasi data mata kuliah ditunjukkan dalam Gambar 4.32.

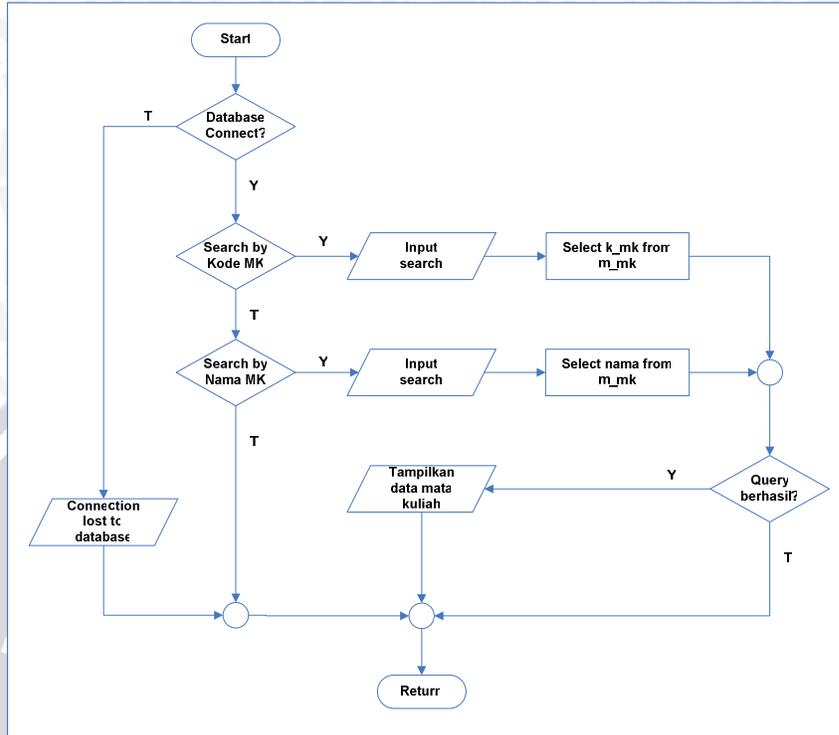


**Gambar 4.32** Diagram Alir Administrasi Data Mata Kuliah  
Sumber: *Perancangan*

Proses yang terjadi pada administrasi data mata kuliah dijelaskan sebagai berikut:

1. Proses menambah data mata kuliah dilakukan dengan memasukkan data mata kuliah baru ke dalam *form* pendataan mata kuliah. Data mata kuliah yang dimasukkan akan disimpan dalam tabel *m\_mk*.
2. Proses mengubah data mata kuliah dilakukan dengan mengubah data mata kuliah yang terdapat dalam *form* pendataan mata kuliah. Data mata kuliah yang diubah akan disimpan dalam tabel *m\_mk*.
3. Proses menghapus data mata kuliah dilakukan dengan menghapus data mata kuliah terdapat dalam *form* pendataan mata kuliah. Data mata kuliah yang dihapus akan dihilangkan dari tabel *m\_mk*.
4. Proses pencarian data mata kuliah merupakan proses untuk mencari data mata kuliah sesuai dengan kehendak administrator.

Diagram alir pencarian data mata kuliah ditunjukkan dalam Gambar 4.33.



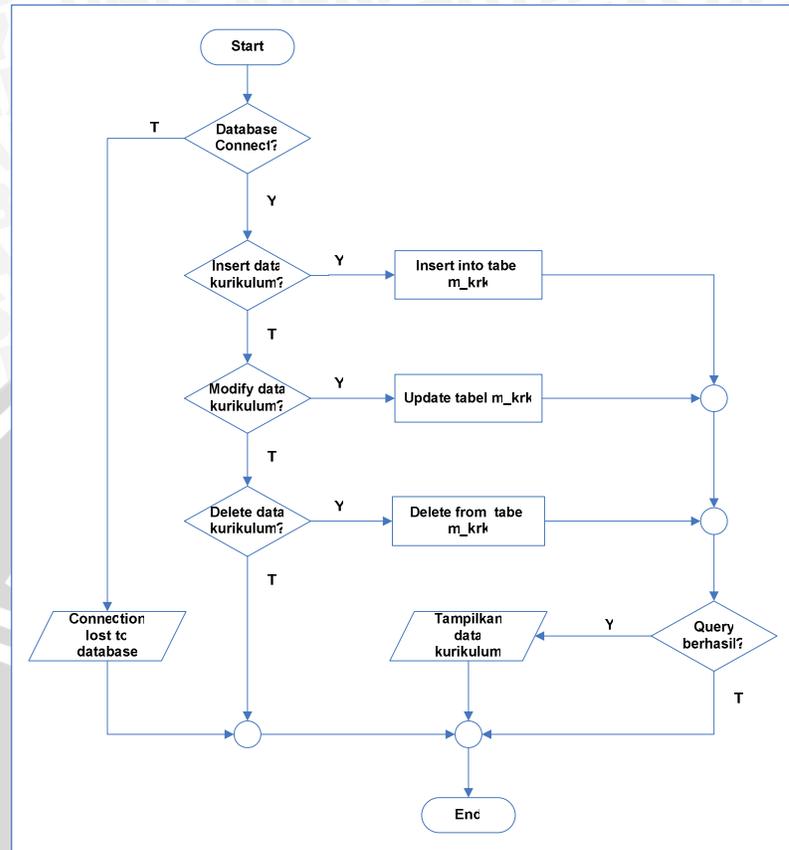
**Gambar 4.33** Diagram Alir Pencarian Data Mata Kuliah  
Sumber: *Perancangan*

Proses pencarian data mata kuliah dipisahkan berdasarkan kode mata kuliah dan nama mata kuliah. Pencarian kode mata kuliah dilakukan dengan membandingkan masukan kode mata kuliah pada *form* mata kuliah dengan kode mata kuliah pada tabel *m\_mk*. Pencarian nama mata kuliah dilakukan dengan membandingkan masukan nama mata kuliah pada *form* pendataan mata kuliah dengan nama mata kuliah pada tabel *m\_mk*. Hasil pencarian data mata kuliah akan ditampilkan pada *form* pendataan mata kuliah.

#### 4.4.3.1.5 Diagram Alir Proses Administrasi Data Kurikulum

Proses administrasi data kurikulum mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data kurikulum pada tabel *m\_krk*.

Diagram alir administrasi data kurikulum ditunjukkan dalam Gambar 4.34.



Gambar 4.34 Diagram Alir Administrasi Data Kurikulum

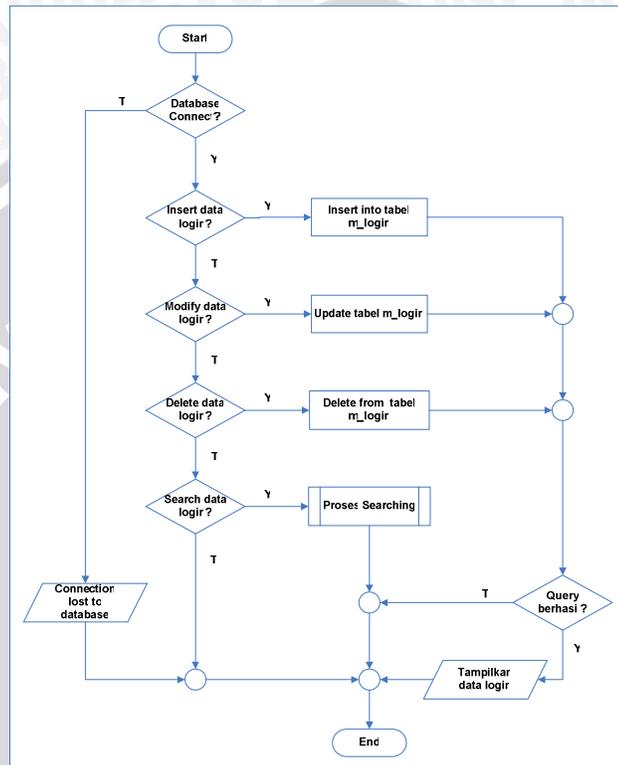
Sumber: Perancangan

Proses yang terjadi pada administrasi data kurikulum dijelaskan sebagai berikut:

1. Proses menambah data kurikulum dilakukan dengan memasukkan data kurikulum baru ke dalam *form* pendataan kurikulum. Data kurikulum yang dimasukkan akan disimpan dalam tabel *m\_krk*.
2. Proses mengubah data kurikulum dilakukan dengan mengubah data kurikulum yang terdapat dalam *form* pendataan kurikulum. Data kurikulum yang diubah akan disimpan dalam tabel *m\_krk*.
3. Proses menghapus data kurikulum dilakukan dengan menghapus data kurikulum terdapat dalam *form* pendataan kurikulum. Data kurikulum yang dihapus akan dihilangkan dari tabel *m\_krk*.

#### 4.4.3.1.6 Diagram Alir Proses Administrasi Data Login

Proses administrasi data *login* mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data *login* pada tabel *m\_login*. Diagram alir administrasi data *login* ditunjukkan dalam Gambar 4.35.



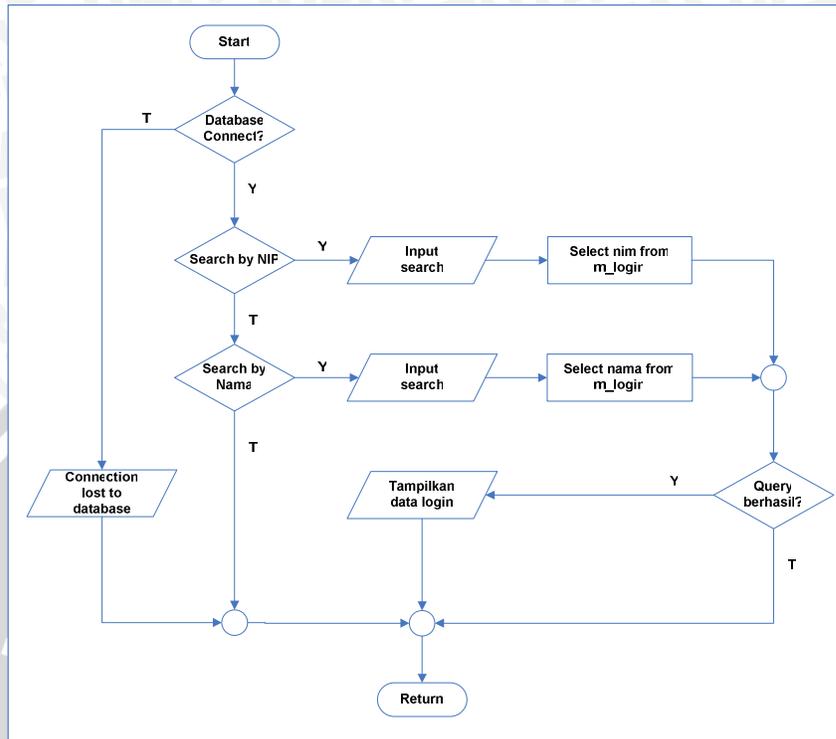
Gambar 4.35 Diagram Alir Administrasi Data Login

Sumber: Perancangan

Proses yang terjadi pada administrasi data *login* dijelaskan sebagai berikut:

1. Proses menambah data *login* dilakukan dengan memasukkan data *login* baru ke dalam *form* pendataan *login*. Data *login* yang dimasukkan akan disimpan dalam tabel *m\_login*.
2. Proses mengubah data *login* dilakukan dengan mengubah data *login* yang terdapat dalam *form* pendataan *login*. Data *login* yang diubah akan disimpan dalam tabel *m\_login*.
3. Proses menghapus data *login* dilakukan dengan menghapus data *login* terdapat dalam *form* pendataan *login*. Data *login* yang dihapus akan dihilangkan dari tabel *m\_login*.
4. Proses pencarian data *login* merupakan proses untuk mencari data *login* sesuai dengan kehendak administrator.

Diagram alir pencarian data *login* ditunjukkan dalam Gambar 4.36.



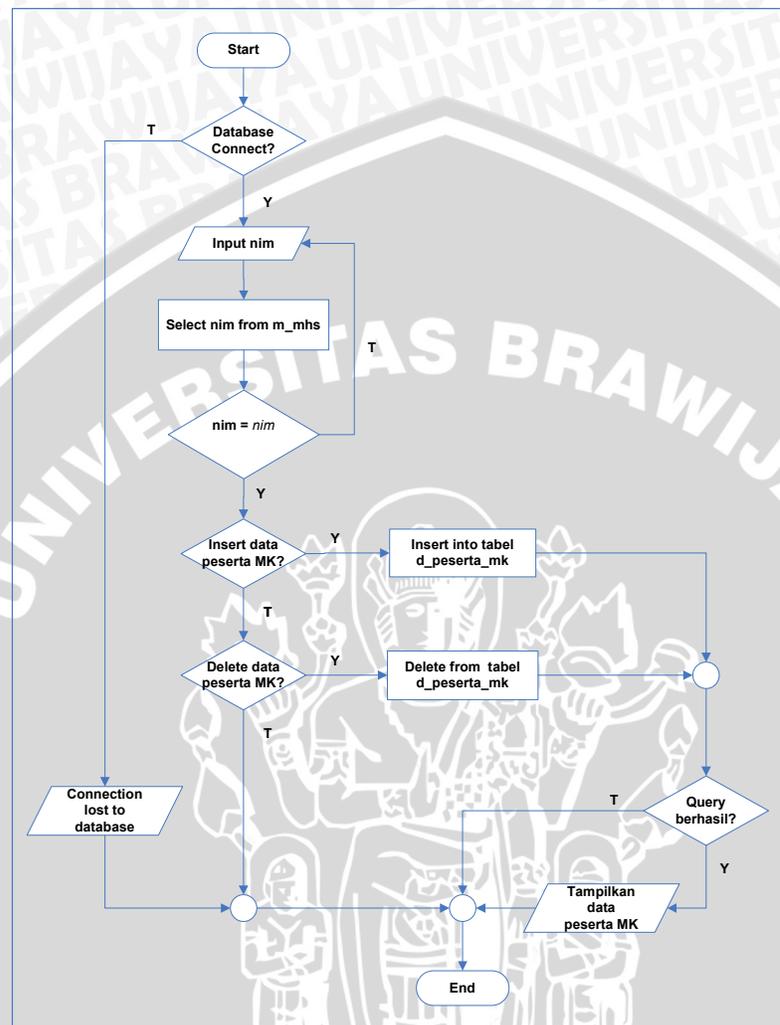
**Gambar 4.36** Diagram Alir Pencarian Data *Login*  
Sumber: Perancangan

Proses pencarian data *login* dipisahkan berdasarkan NIP dosen dan pencarian nama dosen. Pencarian NIP dosen dilakukan dengan membandingkan masukan NIP dosen pada *form* pendataan *login* dengan NIP dosen pada tabel *m\_dosen* yang berelasi dengan tabel *m\_login*. Pencarian nama dosen dilakukan dengan membandingkan masukan nama dosen pada *form* pendataan *login* dengan nama dosen pada tabel *m\_dosen* yang berelasi dengan tabel *m\_login*. Hasil pencarian data *login* akan ditampilkan pada *form* pendataan *login*.

#### 4.4.3.1.7 Diagram Alir Proses Administrasi Data Peserta Mata Kuliah

Proses administrasi data peserta mata kuliah mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data peserta mata kuliah pada tabel *d\_peserta\_mk*.

Diagram alir administrasi data peserta mata kuliah ditunjukkan dalam Gambar 4.37.



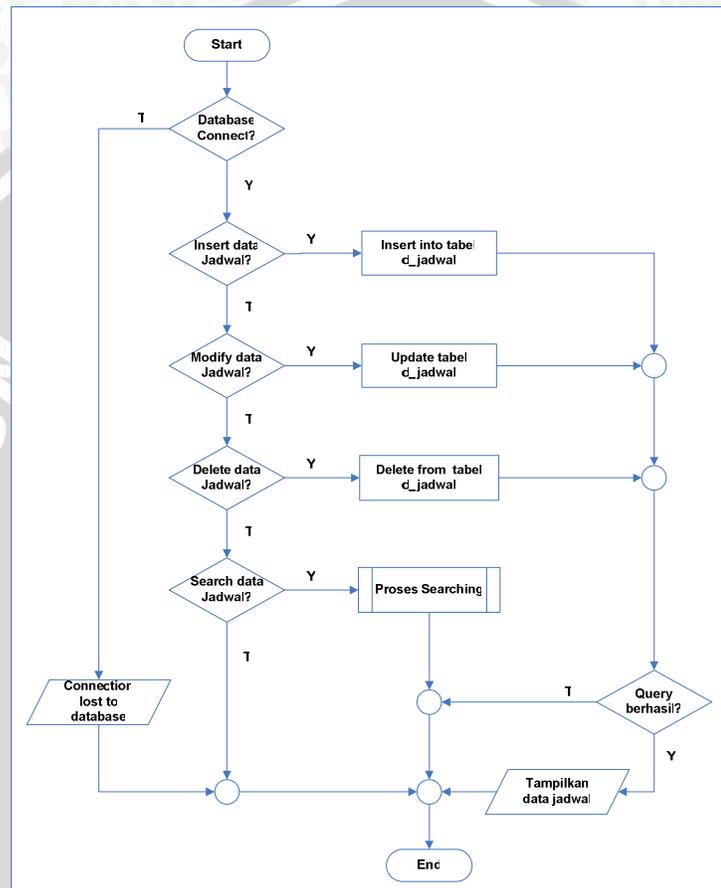
**Gambar 4.37** Diagram Alir Administrasi Data Peserta Mata Kuliah  
Sumber: *Perancangan*

Proses yang terjadi pada administrasi data peserta mata kuliah dijelaskan sebagai berikut:

1. Proses menambah data peserta mata kuliah dilakukan dengan memasukkan data peserta mata kuliah baru ke dalam *form* pendataan peserta mata kuliah. Data peserta mata kuliah yang dimasukkan akan disimpan dalam tabel `d_peserta_mk`.
2. Proses menghapus data peserta mata kuliah dilakukan dengan menghapus data peserta mata kuliah terdapat dalam *form* peserta mata kuliah. Data peserta mata kuliah yang dihapus akan dihilangkan dari tabel `d_peserta_mk`.

#### 4.4.3.1.8 Diagram Alir Proses Administrasi Data Jadwal Mata Kuliah

Proses administrasi data jadwal mata kuliah mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data jadwal mata kuliah pada tabel `d_jadwal`. Diagram alir administrasi data jadwal mata kuliah ditunjukkan dalam Gambar 4.38.



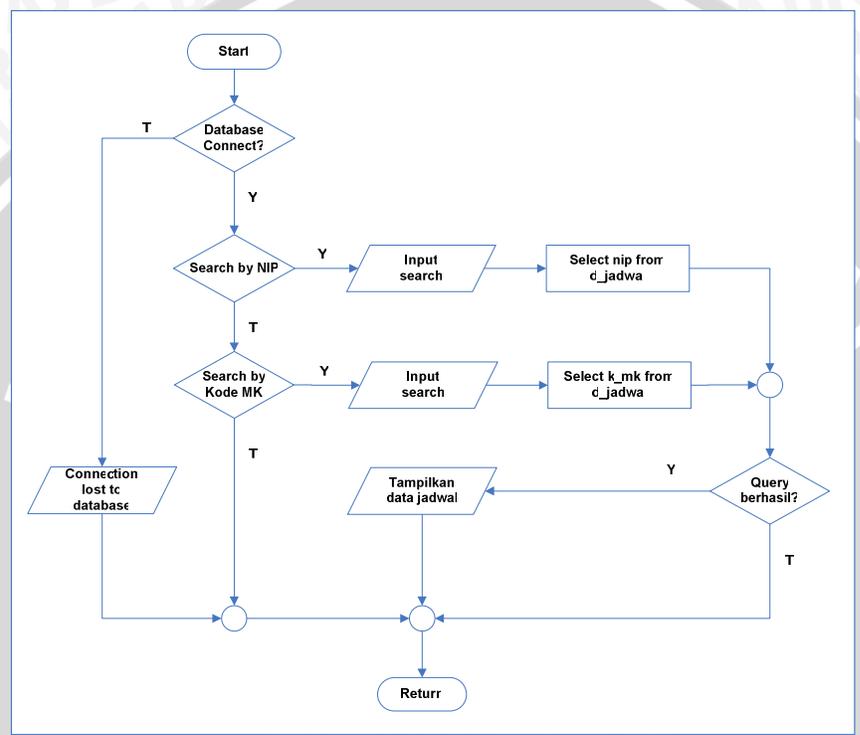
**Gambar 4.38** Diagram Alir Administrasi Data Jadwal Mata Kuliah  
Sumber: *Perancangan*

Proses yang terjadi pada administrasi data jadwal mata kuliah dijelaskan sebagai berikut:

1. Proses menambah data jadwal mata kuliah dilakukan dengan memasukkan data jadwal mata kuliah baru ke dalam *form* pendataan jadwal mata kuliah. Data jadwal mata kuliah yang dimasukkan akan disimpan dalam tabel `d_jadwal`.
2. Proses mengubah data jadwal mata kuliah dilakukan dengan mengubah data jadwal mata kuliah yang terdapat dalam *form* pendataan jadwal mata kuliah. Data jadwal mata kuliah yang diubah akan disimpan dalam tabel `d_jadwal`.

3. Proses menghapus data jadwal mata kuliah dilakukan dengan menghapus data jadwal mata kuliah terdapat dalam *form* pendataan jadwal mata kuliah. Data jadwal mata kuliah yang dihapus akan dihilangkan dari tabel d\_jadwal.
4. Proses pencarian data jadwal mata kuliah merupakan proses untuk mencari data jadwal mata kuliah sesuai dengan kehendak administrator.

Diagram alir pencarian data jadwal mata kuliah ditunjukkan dalam Gambar 4.39.

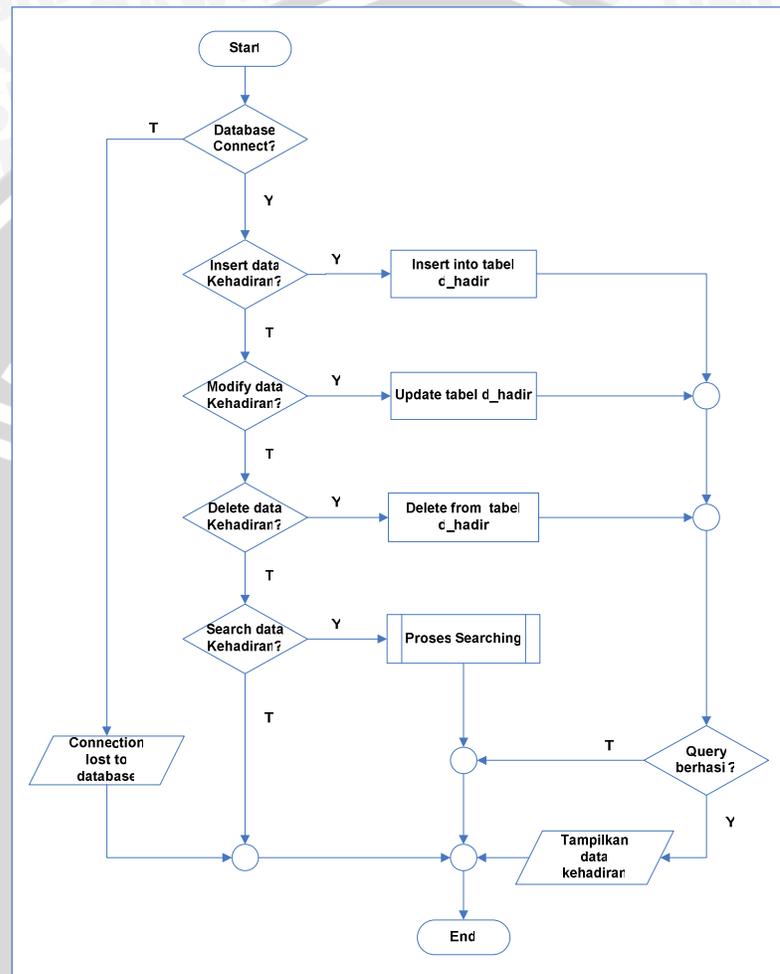


**Gambar 4.39** Diagram Alir Pencarian Jadwal Mata Kuliah  
 Sumber: *Perancangan*

Proses pencarian data jadwal mata kuliah dipisahkan berdasarkan NIP dosen dan kode mata kuliah. Pencarian NIP dosen dilakukan dengan membandingkan masukan NIP dosen pada *form* pendataan jadwal mata kuliah dengan NIP dosen pada tabel d\_jadwal. Pencarian kode mata kuliah dilakukan dengan membandingkan masukan kode mata kuliah pada *form* pendataan jadwal mata kuliah dengan kode mata kuliah pada tabel d\_jadwal. Hasil pencarian data jadwal mata kuliah akan ditampilkan pada *form* pendataan jadwal mata kuliah.

#### 4.4.3.1.9 Diagram Alir Proses Administrasi Data Kehadiran Mahasiswa

Proses administrasi data kehadiran mahasiswa mempunyai beberapa proses, yaitu proses menambah data, mengubah data, menghapus data, dan mencari data kehadiran mahasiswa pada tabel `d_hadir`. Diagram alir administrasi data kehadiran mahasiswa ditunjukkan dalam Gambar 4.40.



**Gambar 4.40** Diagram Alir Administrasi Data Kehadiran Mahasiswa  
Sumber: Perancangan

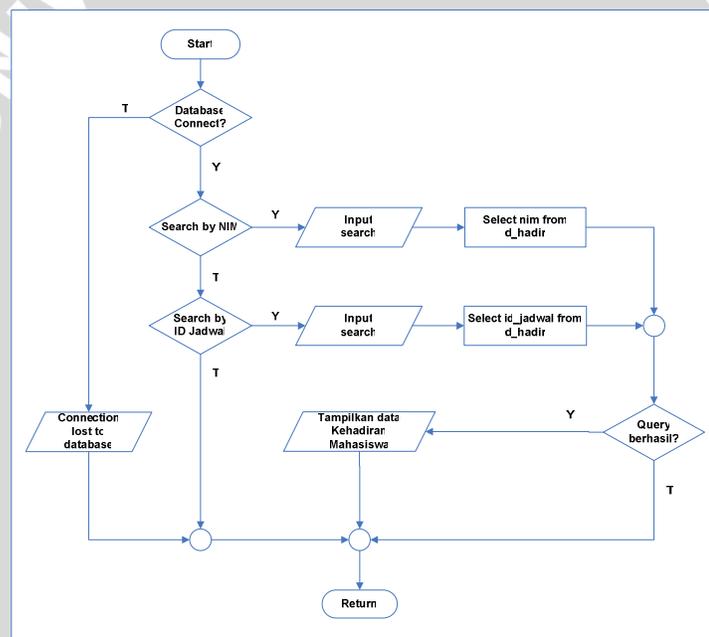
Proses yang terjadi pada administrasi data kehadiran mahasiswa dijelaskan sebagai berikut:

1. Proses menambah data kehadiran mahasiswa dilakukan dengan memasukkan data kehadiran mahasiswa baru ke dalam *form* pendataan kehadiran mahasiswa. Data kehadiran mahasiswa yang dimasukkan akan disimpan dalam tabel `d_hadir`.

2. Proses mengubah data kehadiran mahasiswa dilakukan dengan mengubah data kehadiran mahasiswa yang terdapat dalam *form* pendataan kehadiran mahasiswa. Data kehadiran mahasiswa yang diubah akan disimpan dalam tabel `d_hadir`.
3. Proses menghapus data kehadiran mahasiswa dilakukan dengan menghapus data kehadiran mahasiswa terdapat dalam *form* pendataan kehadiran mahasiswa. Data kehadiran mahasiswa yang dihapus akan dihilangkan dari tabel `d_hadir`.
4. Proses pencarian data kehadiran mahasiswa merupakan proses untuk mencari data kehadiran mahasiswa sesuai dengan kehendak administrator.

Diagram alir pencarian data kehadiran mahasiswa ditunjukkan dalam Gambar

4.41.



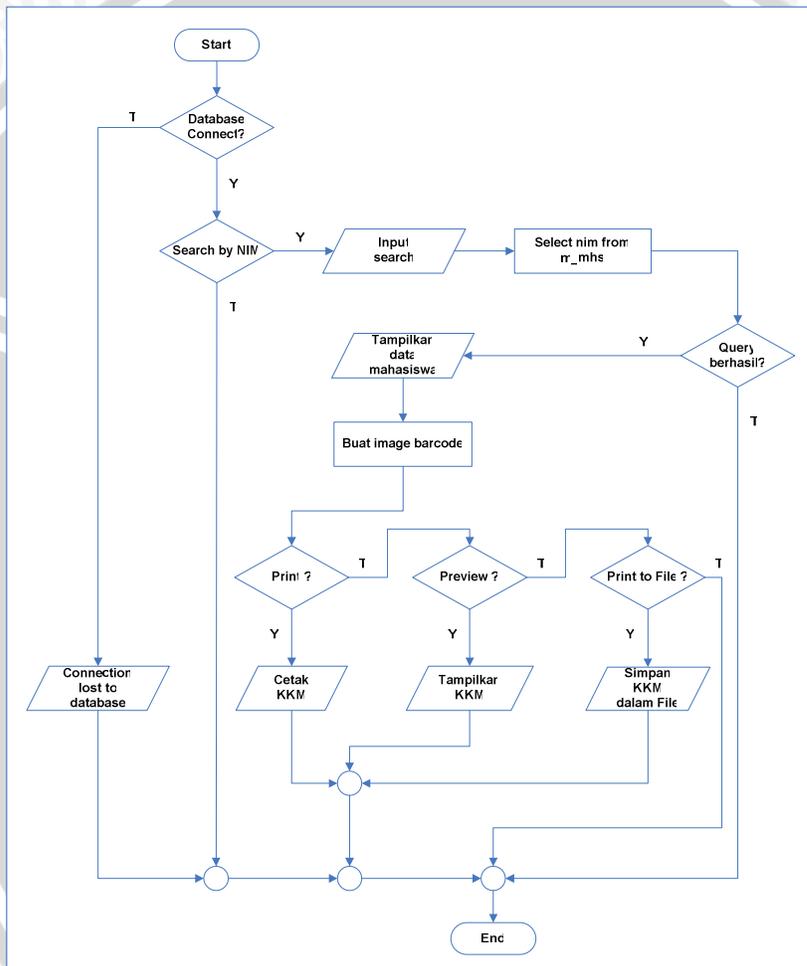
**Gambar 4.41** Diagram Alir Pencarian Data Kehadiran Mahasiswa

Sumber: *Perancangan*

Proses pencarian data kehadiran mahasiswa dipisahkan berdasarkan NIM mahasiswa dan id jadwal. Pencarian NIM mahasiswa dilakukan dengan membandingkan masukan NIM mahasiswa pada *form* pendataan kehadiran mahasiswa dengan NIM mahasiswa pada tabel `d_hadir`. Pencarian kode id jadwal dilakukan dengan membandingkan masukan id jadwal pada *form* pendataan kehadiran mahasiswa dengan id jadwal pada tabel `d_hadir`. Hasil pencarian data kehadiran mahasiswa akan ditampilkan pada *form* pendataan kehadiran mahasiswa.

**4.4.3.1.10 Diagram Alir Proses Pencetakan Kartu Kehadiran Mahasiswa (KKM)**

Proses pencetakan KKM membutuhkan masukan data mahasiswa yang diperoleh dari masukan NIM mahasiswa. NIM mahasiswa yang diperoleh dari tabel *m\_mhs* akan ditampilkan pada *form* pencetakan KKM. Administrator akan diberikan pilihan, yaitu mencetak KKM (*print*), melihat tampilan KKM (*preview*), atau menyimpan KKM dalam *file*. Diagram alir pencetakan KKM ditunjukkan dalam Gambar 4.42.

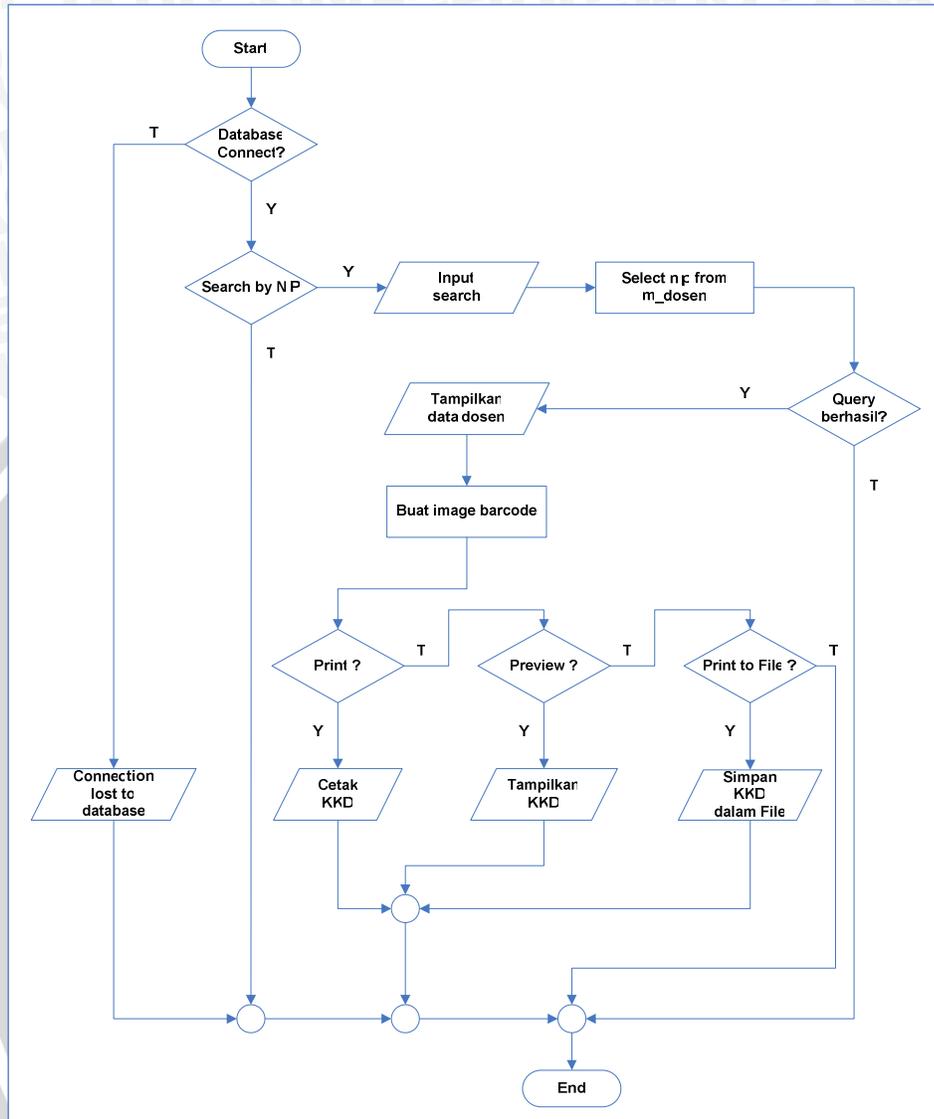


**Gambar 4.42** Diagram Alir Pencetakan Kartu Kehadiran Mahasiswa  
 Sumber: Perancangan

**4.4.3.1.11 Diagram Alir Proses Pencetakan Kartu Kehadiran Dosen (KKD)**

Proses pencetakan KKD membutuhkan masukan data dosen yang diperoleh dari masukan NIP dosen. NIP dosen yang diperoleh dari tabel *m\_dosen* akan ditampilkan pada *form* pencetakan KKD. Administrator akan diberikan pilihan, yaitu mencetak KKM (*print*), melihat tampilan KKM (*preview*), atau menyimpan KKM dalam *file*.

Diagram alir pencetakan KKD ditunjukkan dalam Gambar 4.43.

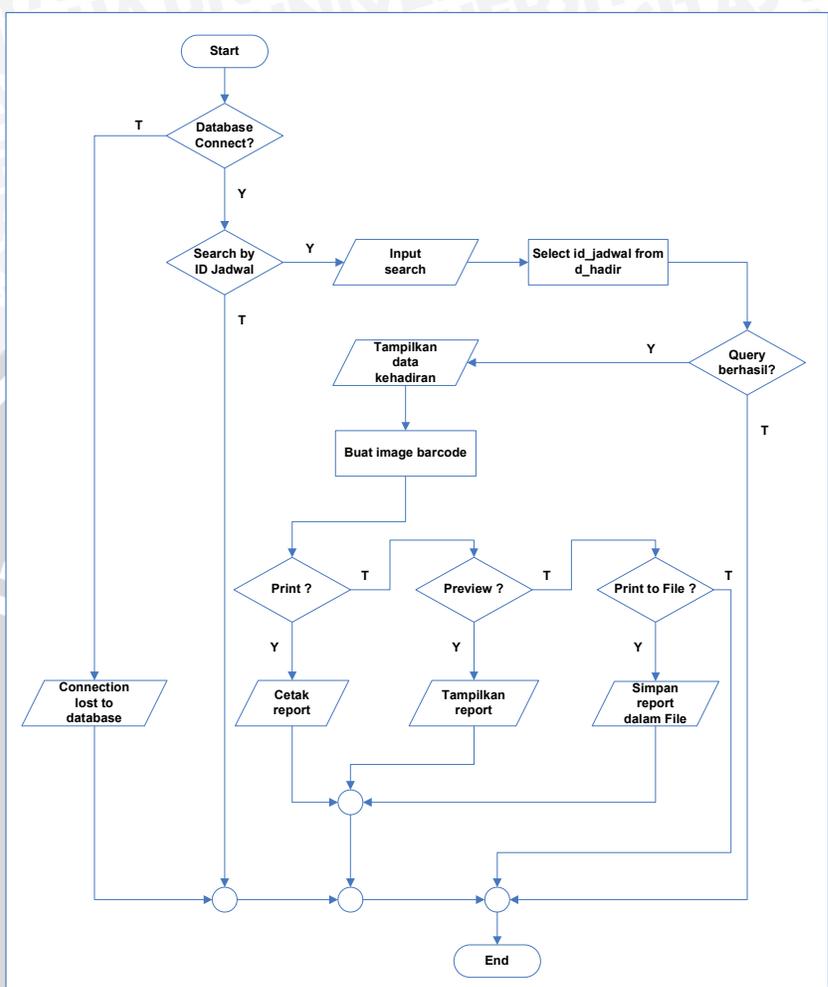


Gambar 4.43 Diagram Alir Pencetakan Kartu Kehadiran Dosen  
 Sumber: Perancangan

#### 4.4.3.1.12 Diagram Alir Proses Pencetakan Report Kehadiran Mahasiswa

Proses pencetakan *report* kehadiran mahasiswa membutuhkan data kehadiran mahasiswa yang diperoleh dari masukan kode mata kuliah. Kode mata kuliah akan ditampilkan pada *form* pencetakan *report* kehadiran mahasiswa. Administrator akan diberikan pilihan, yaitu mencetak *report* (*print*), melihat tampilan *report* (*preview*), atau menyimpan *report* dalam *file*.

Diagram alir pencetakan *report* kehadiran mahasiswa ditunjukkan dalam Gambar 4.44.

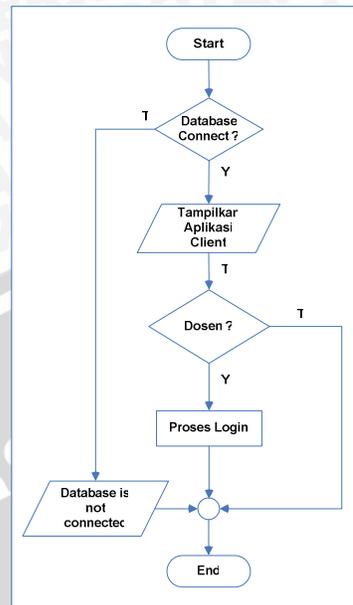


Gambar 4.44 Diagram Alir Pencetakan *Report* Kehadiran Mahasiswa  
 Sumber: Perancangan

#### 4.4.3.2 Diagram Alir Aplikasi *Client*

Diagram alir aplikasi *client* menjelaskan proses yang terjadi pada aplikasi *client*. Proses yang terjadi pada aplikasi *client* sesuai dengan yang dijelaskan pada sub-bab 4.4.1.2. Aplikasi *client* akan mengecek koneksi aplikasi terhadap basis data *server*, bila tidak terhubung maka pesan peringatan akan ditampilkan. Aplikasi *client* akan ditampilkan setelah melalui pengecekan koneksi basis data *server*. Proses *login* untuk mengecek hak akses dari pengguna yang ingin mengakses aplikasi *client*. Pesan kesalahan akan ditampilkan bila pengguna tidak mempunyai hak akses terhadap aplikasi *client*.

Diagram alir aplikasi *client* ditunjukkan dalam Gambar 4.45.



**Gambar 4.45** Diagram Alir Aplikasi *Client*  
Sumber: *Perancangan*

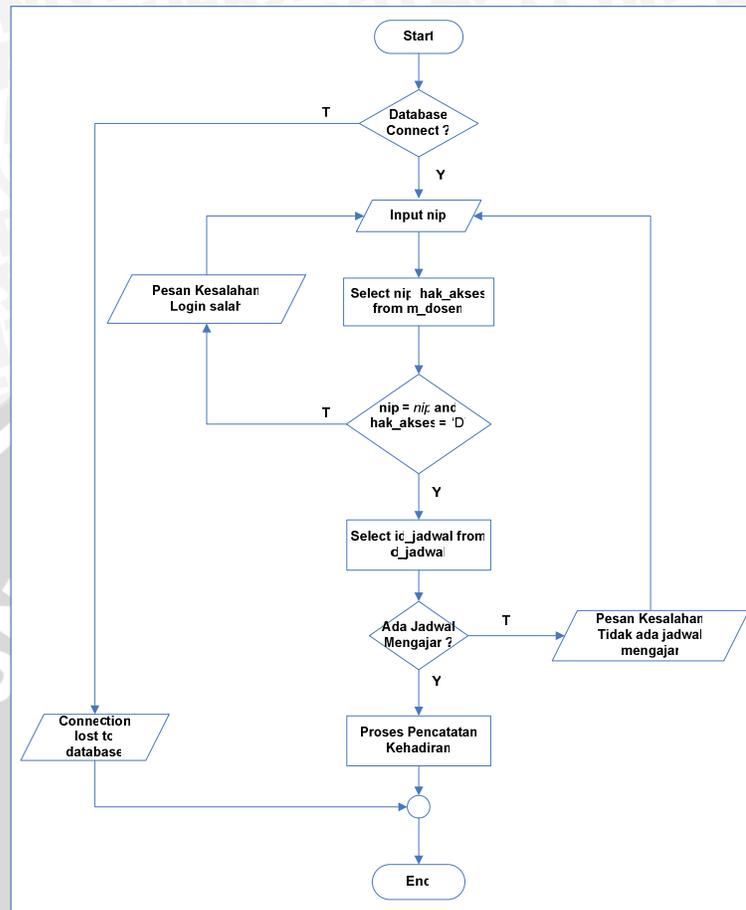
Proses pertama yang terjadi setelah menjalankan *client.exe* adalah pengecekan koneksi pada basis data. Aplikasi *client* menampilkan pesan kesalahan bila *server* basis data tidak berjalan. *User*, dalam hal ini dosen adalah pengguna yang mempunyai hak akses untuk aplikasi *client*. Dosen melakukan proses *login* untuk menjalankan aplikasi *client*.

#### 4.4.3.2.1 Diagram Alir Proses *Login User*

Proses pertama adalah memasukkan NIP dosen melalui pembacaan KKD oleh *barcode scanner* untuk dibandingkan dengan *field* NIP di tabel *m\_dosen*. NIP dosen yang sesuai pada tabel *m\_dosen* akan menampilkan status *login* dosen pada aplikasi *client*. Pesan kesalahan akan ditampilkan bila tidak ada kesesuaian antara data masukan dengan data pada tabel *m\_dosen*.

Proses *login user* juga menentukan pemilihan jadwal mata kuliah. Pemilihan jadwal mata kuliah dilakukan bila pengisian NIP dosen bernilai benar. Pemilihan jadwal mata kuliah berdasarkan hari dan jam mengajar dosen pada tabel *d\_jadwal* yang dibandingkan dengan hari dan jam saat proses *login user* dijalankan. Dosen dapat melakukan proses *login* antara awal jam mengajar sampai jam mengajar berakhir. Proses pencatatan kehadiran mahasiswa berjalan setelah proses *login user* dan proses pemilihan jadwal mata kuliah selesai dilaksanakan.

Diagram alir *login client* ditunjukkan dalam Gambar 4.46.

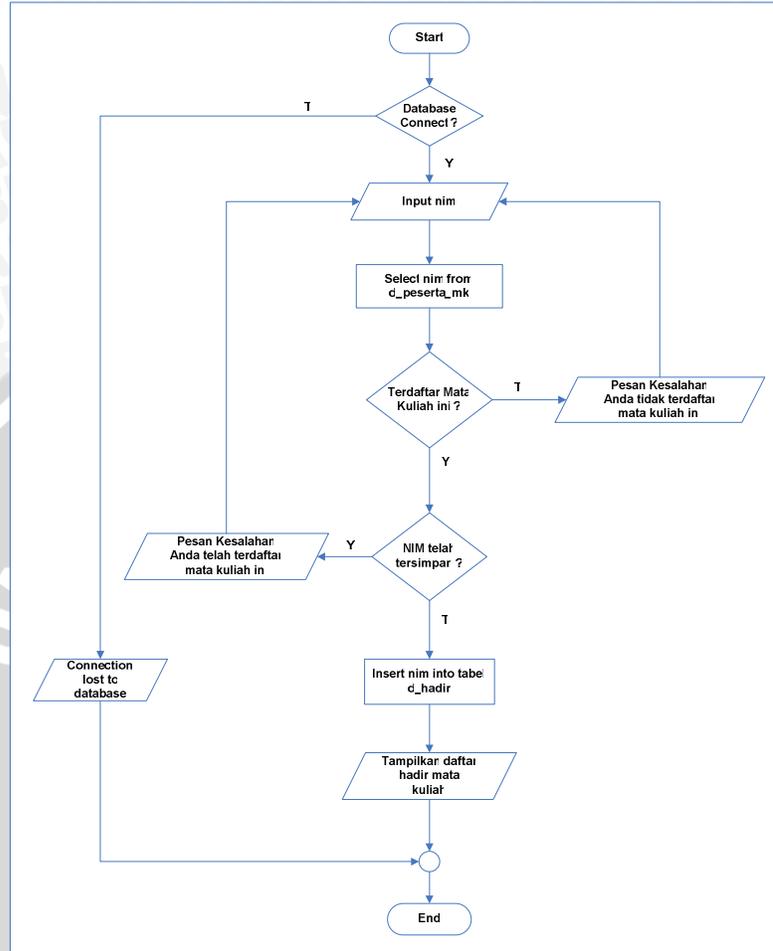


Gambar 4.46 Diagram Alir *Login Client*

Sumber: *Perancangan*

Proses pencatatan kehadiran mahasiswa merupakan proses menyimpan data kehadiran mahasiswa sesuai dengan jadwal mata kuliah yang diikuti oleh mahasiswa tersebut. Proses menyimpan data kehadiran mahasiswa berdasarkan waktu yang berjalan pada aplikasi *server* untuk menjaga integritas data kehadiran mahasiswa. Proses pencatatan kehadiran mahasiswa membutuhkan masukan NIM berbentuk *barcode* dari KKM yang dimiliki oleh mahasiswa. Mahasiswa yang telah terdaftar dalam peserta mata kuliah yang sedang berlangsung akan disimpan pada tabel *d\_hadir* dan ditampilkan dalam *form* kehadiran mahasiswa. Mahasiswa dapat mendaftarkan kehadirannya selama dosen telah *login* dan jam sesuai dengan jadwal mata kuliah yang diikuti belum berakhir. Pesan kesalahan ditampilkan bila mahasiswa tidak terdaftar dalam mata kuliah yang sedang berlangsung. Pesan kesalahan juga ditampilkan bila NIM mahasiswa telah tersimpan dalam daftar hadir mahasiswa dalam satu pertemuan mata kuliah.

Diagram alir pencatatan kehadiran mahasiswa ditunjukkan dalam Gambar 4.47.

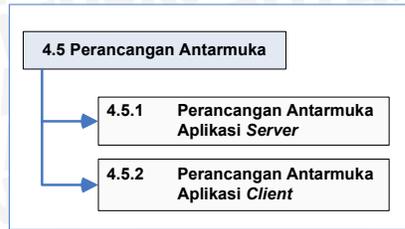


Gambar 4.47 Diagram Alir Pencatatan Kehadiran Mahasiswa  
 Sumber: Perancangan

#### 4.5 Perancangan Antarmuka Sistem Informasi Kehadiran Mahasiswa

Perancangan antarmuka Sistem Informasi Kehadiran Mahasiswa terdiri dari dua bagian, yaitu perancangan aplikasi *server* dan perancangan aplikasi *client*. Perancangan aplikasi *server* dan aplikasi *client* dalam Sistem Informasi Kehadiran Mahasiswa (SIKMA) disesuaikan dengan kebutuhan pengguna berdasarkan DFD, STD dan diagram alir yang telah dijelaskan pada sub-bab 4.4.1 sampai sub-bab 4.4.3.

Diagram pohon perancangan SIKMA ditunjukkan dalam Gambar 4.48.



**Gambar 4.48** Diagram Pohon Perancangan Antarmuka  
Sumber: *Perancangan*

### 4.5.1 Perancangan Antarmuka Aplikasi *Server*

Perancangan antarmuka aplikasi *server* dirancang untuk administrasi data dan pencetakan kartu kehadiran serta *report* kehadiran mahasiswa pada basis data kehadiran. Administrasi data kehadiran mahasiswa meliputi data mahasiswa, data dosen, data mata kuliah, data kurikulum, data *login*, data peserta mata kuliah, data jadwal mata kuliah, data kehadiran mahasiswa. Pencetakan kartu kehadiran meliputi kartu kehadiran mahasiswa dan kartu kehadiran dosen.

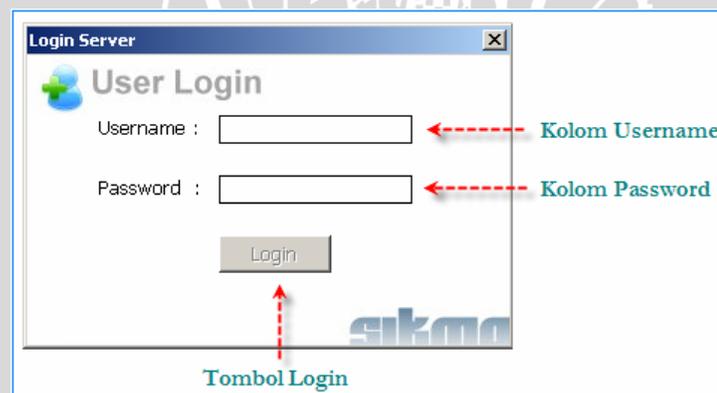
Hak untuk mengakses aplikasi *server* hanya dimiliki oleh administrator, hal ini untuk menjaga penyalahgunaan hak akses dari pihak yang tidak mempunyai wewenang mengakses aplikasi *server*. Pembatasan hak akses dilakukan dengan proses *login* untuk mengaktifkan aplikasi *server*. Perancangan antarmuka aplikasi *server* ditunjukkan dalam Gambar 4.49.



**Gambar 4.49** Antarmuka Aplikasi *Server* sebelum *Login*  
Sumber: *Perancangan*

#### 4.5.1.1 Perancangan Antarmuka *Login Server*

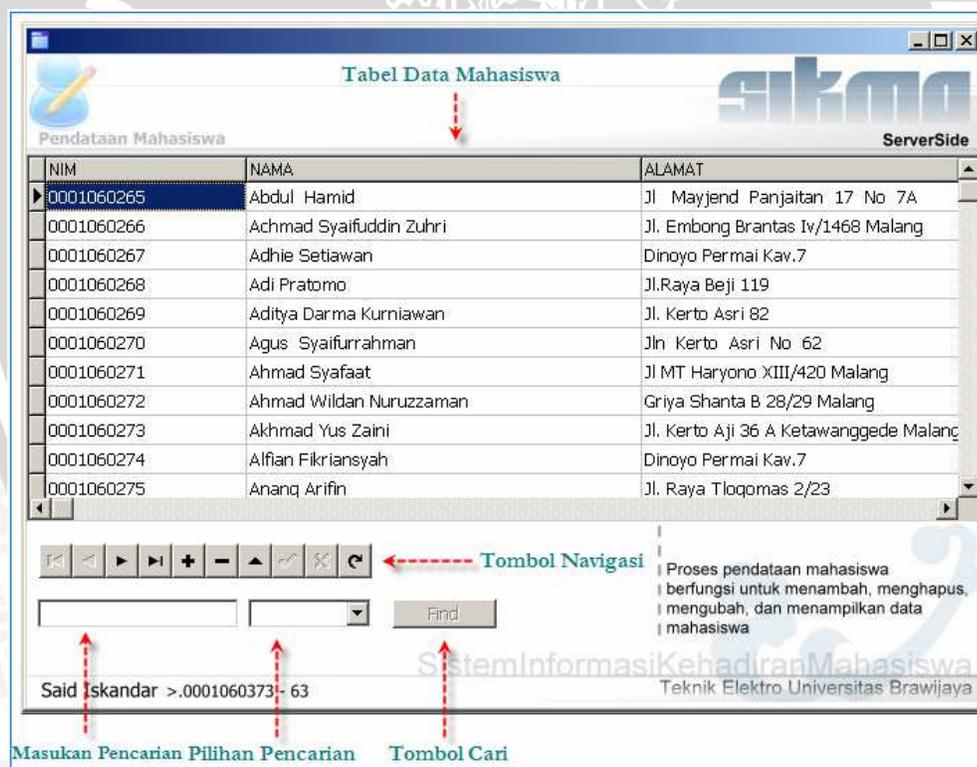
Perancangan antarmuka proses *login server* ditujukan untuk membatasi akses Sistem Informasi Kehadiran Mahasiswa dari pengguna yang tidak mempunyai hak akses terhadap aplikasi *server*. Perancangan antarmuka *login server* terdiri dari kolom *username* dan *password*. Administrator yang menggunakan aplikasi *server* pada Sistem Informasi Kehadiran Mahasiswa harus melalui proses otentifikasi *login server*. Proses otentifikasi *login server* merupakan proses pengecekan masukan *username* dan *password*, dan hak akses dibandingkan dengan yang terdapat pada tabel *m\_login* di basis data kehadiran. Kolom *username* digunakan untuk mengisi nama *login* administrator sesuai dengan tabel *m\_login*. Kolom *password* digunakan untuk mengisi *password* administrator sesuai dengan tabel *m\_login*. Tombol *login* digunakan untuk memproses *username* dan *password* yang diisikan pada kolom *username* dan *password* untuk disesuaikan dengan *username* dan *password* pada tabel *m\_login*. Pesan peringatan akan ditampilkan bila *username* dan *password* tidak sesuai dengan *username* dan *password* pada tabel *m\_login*. Aplikasi *server* akan tertutup bila administrator melakukan kesalahan dalam proses login lebih dari tiga kali. Perancangan antarmuka *login server* ditunjukkan dalam Gambar 4.50.



**Gambar 4.50** Antarmuka *Login Server*  
Sumber: Perancangan

#### 4.5.1.2 Perancangan Antarmuka Administrasi Data Mahasiswa

Perancangan antarmuka administrasi data mahasiswa disesuaikan dengan *data flow diagram level 2* dari proses 2.1. Administrasi data mahasiswa berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data mahasiswa pada tabel *m\_mhs*. Tabel data mahasiswa berisi keterangan data mahasiswa yang terdiri dari nim, nama, dan alamat. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data mahasiswa ditunjukkan dalam Gambar 4.51.

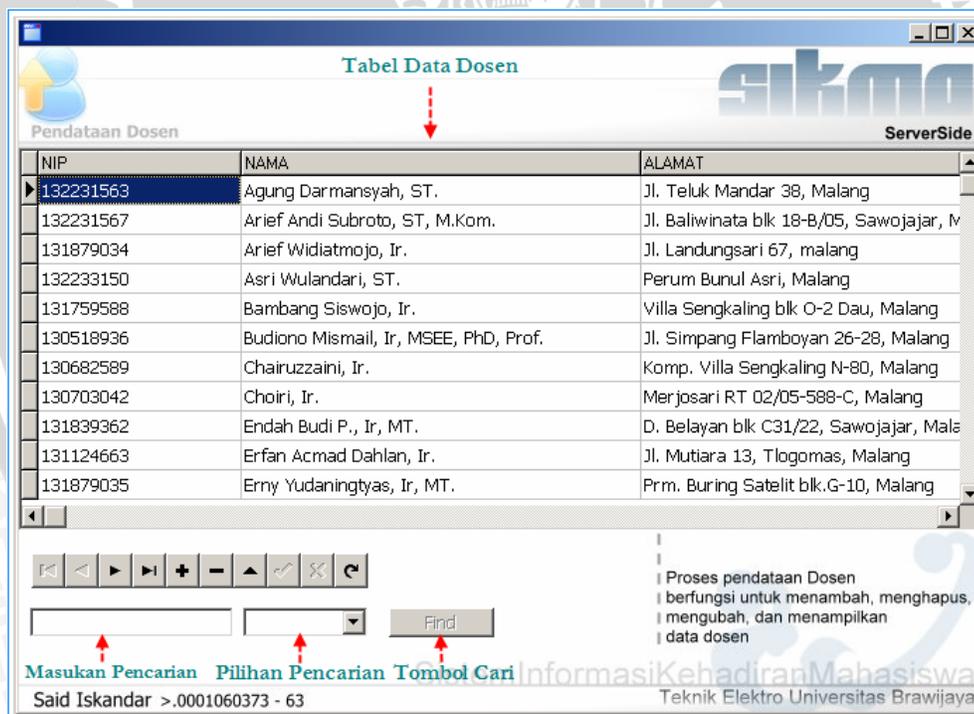


Gambar 4.51 Antarmuka Administrasi Data Mahasiswa

Sumber: Perancangan

#### 4.5.1.3 Perancangan Antarmuka Administrasi Data Dosen

Perancangan antarmuka administrasi data dosen disesuaikan dengan *data flow diagram level 2* dari proses 2.2. Administrasi data dosen berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data dosen pada tabel *m\_dosen*. Tabel data dosen berisi keterangan data dosen yang terdiri dari nip, nama, alamat, dan nomor telepon. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data dosen ditunjukkan dalam Gambar 4.52.

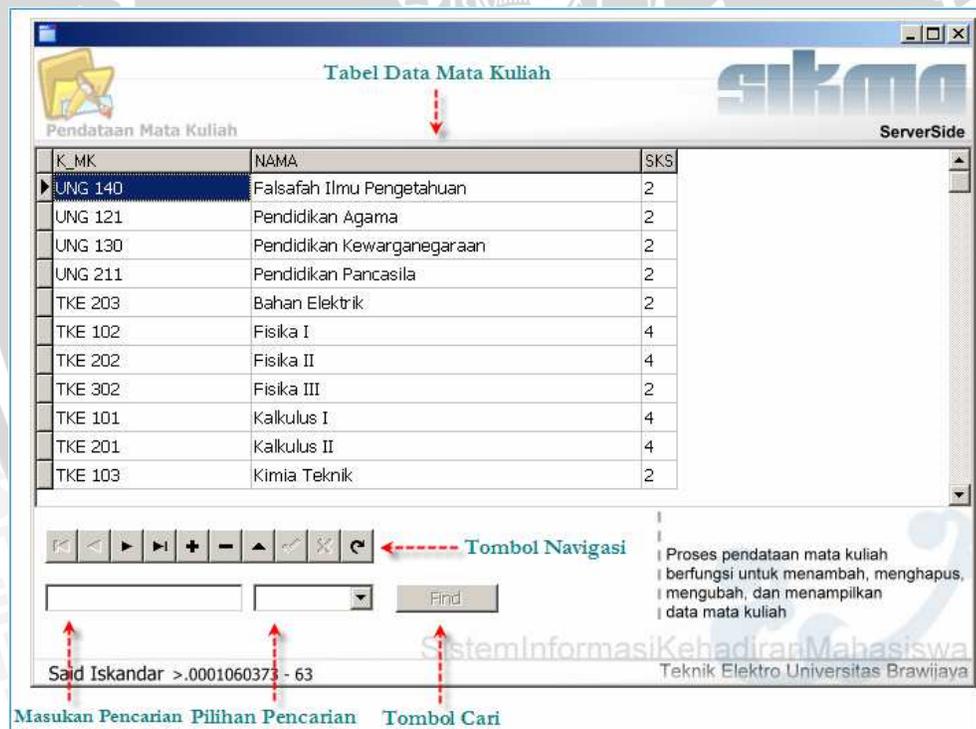


Gambar 4.52 Antarmuka Administrasi Data Dosen

Sumber: Perancangan

#### 4.5.1.4 Perancangan Antarmuka Administrasi Data Mata Kuliah

Perancangan antarmuka administrasi data mata kuliah disesuaikan dengan *data flow diagram level 2* dari proses 2.3. Administrasi data mata kuliah berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data mata kuliah pada tabel m\_mk. Tabel data mata kuliah berisi keterangan data mata kuliah yang terdiri dari kode mata kuliah, nama, dan sks. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data mata kuliah ditunjukkan dalam Gambar 4.53.

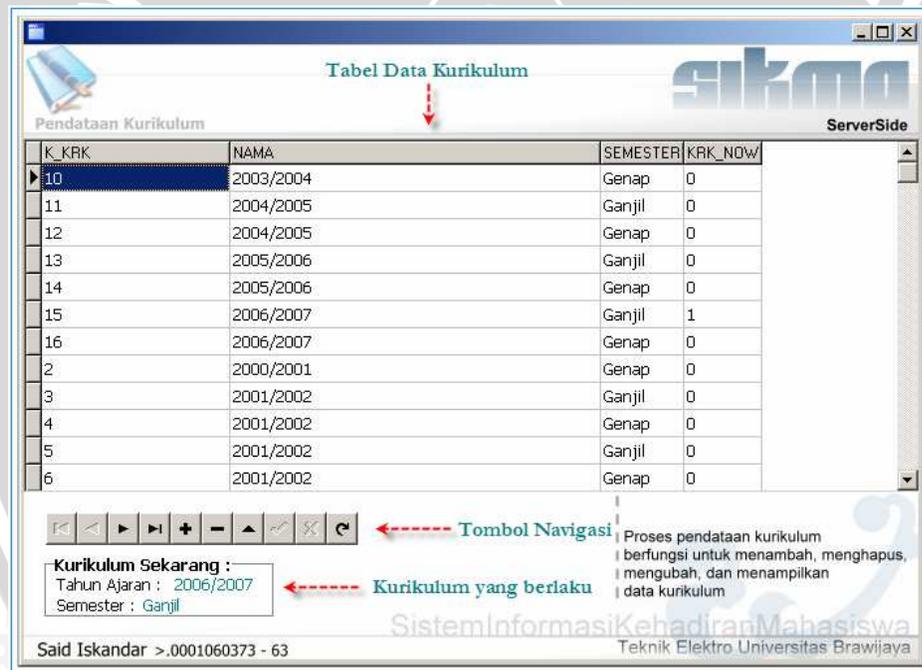


Gambar 4.53 Antarmuka Administrasi Data Mata Kuliah

Sumber: Perancangan

#### 4.5.1.5 Perancangan Antarmuka Administrasi Data Kurikulum

Perancangan antarmuka administrasi data kurikulum disesuaikan dengan *data flow diagram level 2* dari proses 2.4. Administrasi data kurikulum berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data kurikulum pada tabel *m\_krk*. Tabel data kurikulum berisi keterangan data kurikulum yang terdiri dari kode kurikulum, nama, semester, dan kurikulum sekarang. Kurikulum sekarang menjelaskan kurikulum yang berlaku saat aplikasi *server* dijalankan. Kurikulum sekarang terdiri dari tahun ajaran dan semester. Tabel data kurikulum dengan kolom kurikulum sekarang dengan nilai data satu menunjukkan kurikulum yang berlaku sekarang. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data kurikulum ditunjukkan dalam Gambar 4.54.

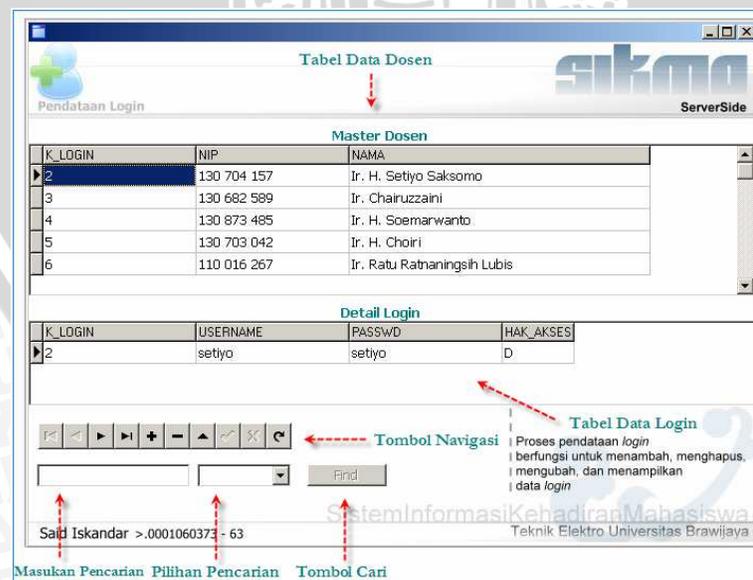


Gambar 4.54 Antarmuka Administrasi Data Kurikulum

Sumber: Perancangan

#### 4.5.1.6 Perancangan Antarmuka Administrasi Data Login

Perancangan antarmuka administrasi data *login* disesuaikan dengan *data flow diagram level 2* dari proses 2.5. Administrasi data *login* berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data *login* pada tabel *m\_login*. Perancangan antarmuka administrasi data *login* terdiri dari dua tabel data, yaitu tabel data dosen dan tabel data *login*. Tabel data *login* berisi keterangan data *login* yang terdiri dari kode *login*, *username*, *password*, dan hak akses. Hak akses dalam proses *login* dibagi dua, yaitu hak akses sebagai administrator dan hak akses sebagai dosen. Administrator mempunyai hak untuk mengakses aplikasi *server*. Hak akses sebagai administrator ditunjukkan dengan huruf 'A' pada tabel *m\_login*. Dosen mempunyai hak untuk mengakses aplikasi *client*. Hak akses sebagai dosen ditunjukkan dengan huruf 'D' pada tabel *m\_login*. Tabel data dosen berisi kode *login* dosen, nip, dan nama dosen. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data *login* ditunjukkan dalam Gambar 4.55.

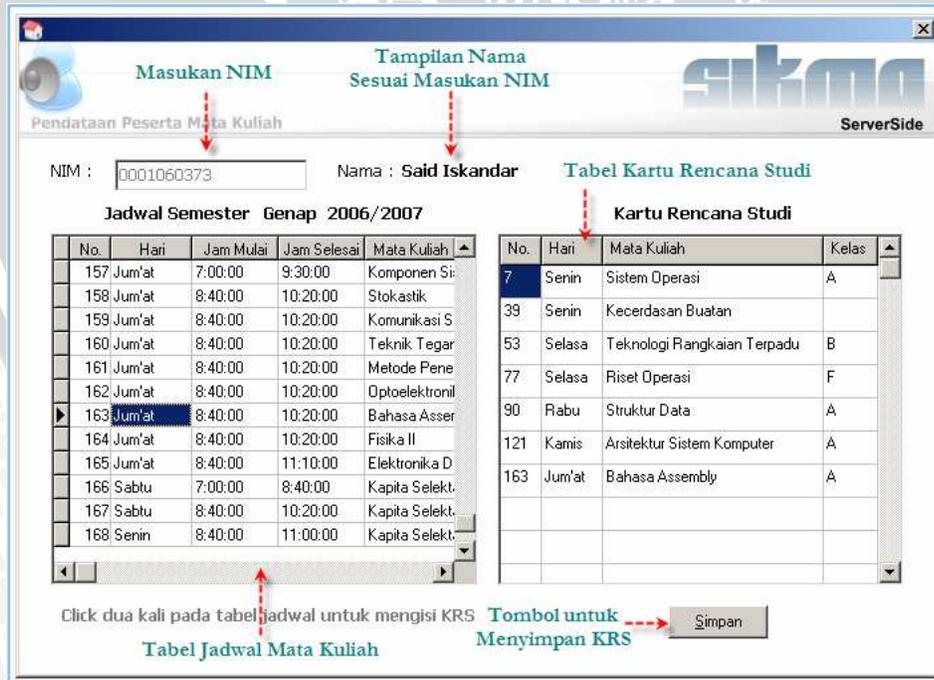


Gambar 4.55 Antarmuka Administrasi Data Login

Sumber: Perancangan

#### 4.5.1.7 Perancangan Antarmuka Administrasi Data Peserta Mata Kuliah

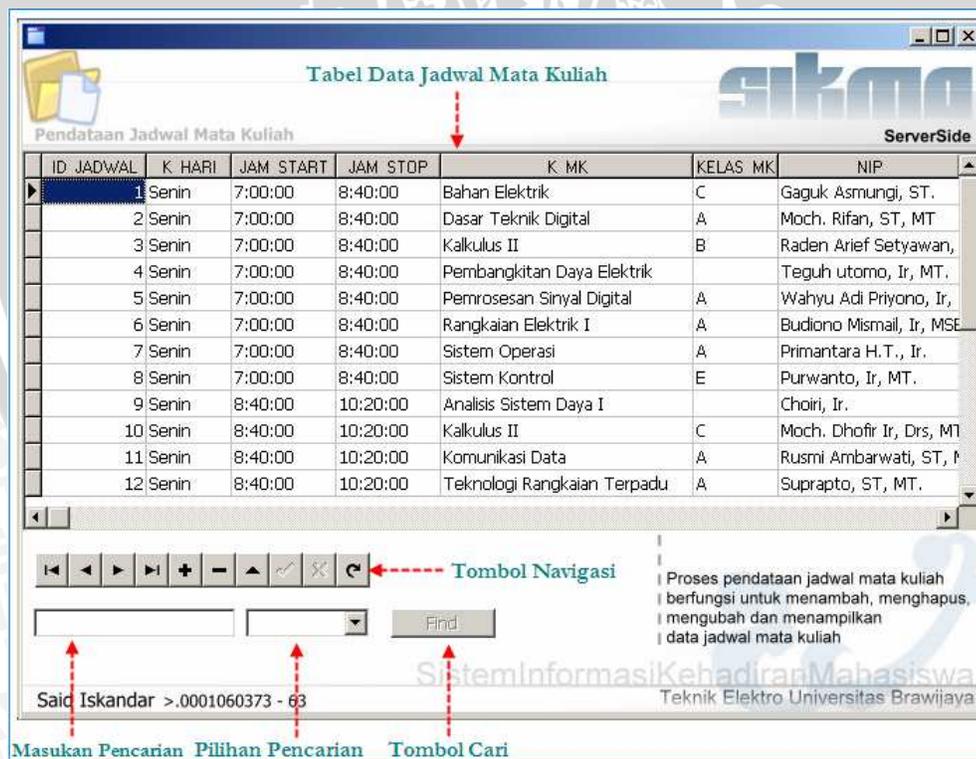
Perancangan antarmuka administrasi data peserta mata kuliah disesuaikan dengan *data flow diagram level 2* dari proses 2.6. Administrasi data peserta mata kuliah berfungsi untuk menambah, menghapus, dan menampilkan data peserta mata kuliah pada tabel `d_peserta_mk`. Perancangan antarmuka administrasi peserta mata kuliah terdiri dari dua tabel data, yaitu tabel data Kartu Rencana Studi (KRS) dan tabel data jadwal. Tabel data KRS berisi keterangan data jadwal mata kuliah yang terdiri dari id jadwal, hari, mata kuliah dan kelas. Tabel data jadwal mata kuliah berisi keterangan jadwal mata kuliah yang terdiri dari id jadwal, hari, jam mulai, jam selesai, kode mata kuliah, nama mata kuliah, dan nama dosen pengajar. Masukan NIM digunakan untuk mengisikan NIM yang digunakan dalam proses pengisian KRS. Tampilan nama digunakan untuk menampilkan nama mahasiswa sesuai dengan masukan NIM. Tombol simpan digunakan untuk menyimpan KRS ke dalam basis data kehadiran. Administrator dapat mengisi KRS dengan memilih jadwal mata kuliah setelah memasukkan NIM. Perancangan antarmuka administrasi data peserta mata kuliah ditunjukkan dalam Gambar 4.56.



**Gambar 4.56** Antarmuka Administrasi Data Peserta Mata Kuliah  
 Sumber: Perancangan

#### 4.5.1.8 Perancangan Antarmuka Administrasi Data Jadwal Mata Kuliah

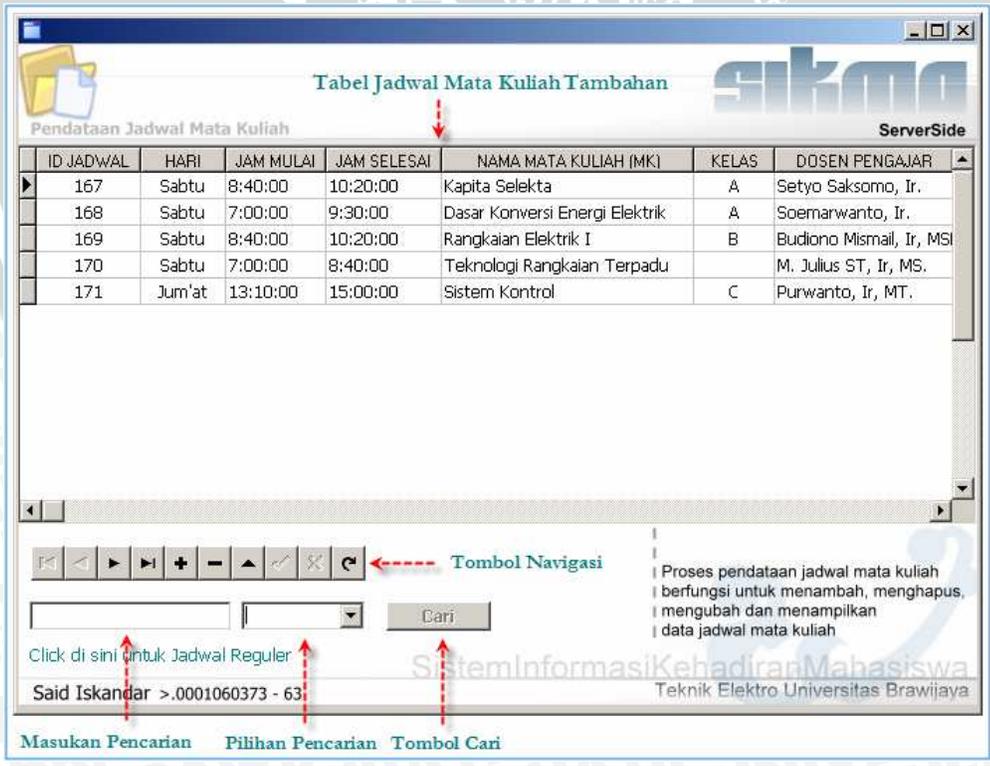
Perancangan antarmuka administrasi data jadwal mata kuliah disesuaikan dengan *data flow diagram level 2* dari proses 2.7. Administrasi data jadwal mata kuliah berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data jadwal mata kuliah pada tabel `d_jadwal`. Tabel data jadwal mata kuliah berisi keterangan data jadwal mata kuliah yang terdiri dari id jadwal, kode hari, kelas mata kuliah, nip dosen pengajar, jam mulai, jam selesai, dan kode kurikulum. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data jadwal mata kuliah ditunjukkan dalam Gambar 4.57.



**Gambar 4.57** Antarmuka Administrasi Data Jadwal Mata Kuliah  
Sumber: Perancangan

#### 4.5.1.8.1 Perancangan Antarmuka Administrasi Data Jadwal Mata Kuliah Tambahan

Perancangan antarmuka administrasi data jadwal mata kuliah tambahan disesuaikan dengan *data flow diagram level 2* dari proses 2.7. Administrasi data jadwal mata kuliah tambahan berfungsi untuk menambah, menghapus, mengubah, dan menampilkan data jadwal mata kuliah tambahan pada tabel `d_jadwal`. Tabel data jadwal mata kuliah tambahan berisi keterangan data jadwal mata kuliah tambahan yang terdiri dari id jadwal, kode hari, kelas mata kuliah, nip dosen pengajar, jam mulai, jam selesai, dan kode kurikulum. Masukan pencarian digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencarian. Pilihan pencarian digunakan untuk memilih kriteria pencarian. Tombol cari digunakan untuk memulai proses pencarian setelah memilih kriteria pencarian dan mengisikan kata kunci yang ingin dicari. Administrator dapat mengisikan kata kunci pencarian setelah memilih kriteria pencarian. Tombol navigasi digunakan untuk menambah, mengubah, menghapus, dan menyimpan data pada basis data kehadiran. Perancangan antarmuka administrasi data jadwal mata kuliah tambahan ditunjukkan dalam Gambar 4.58.

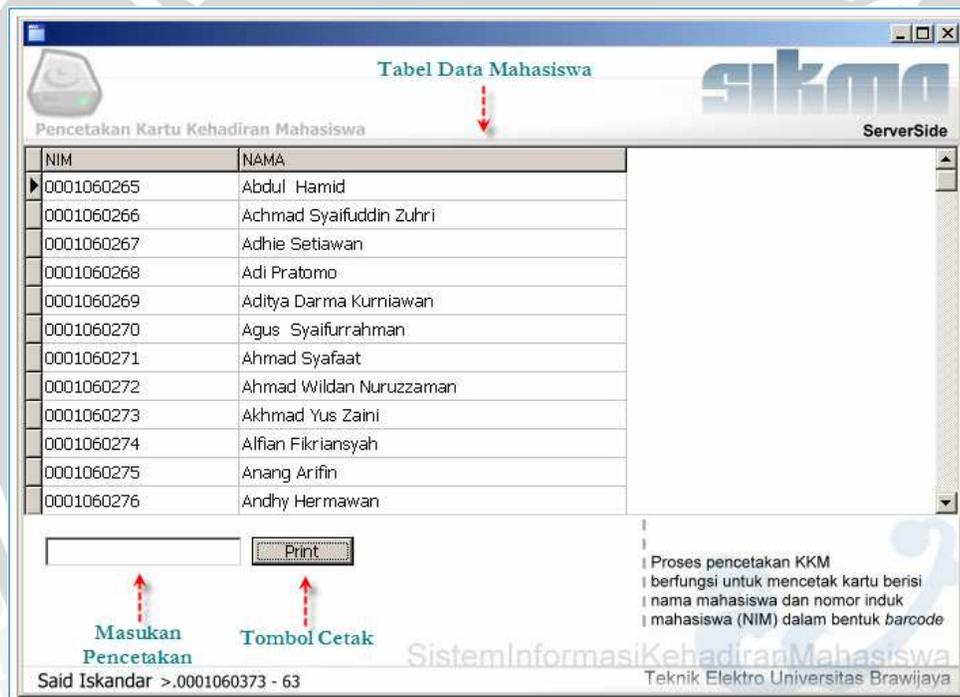


Gambar 4.58 Antarmuka Administrasi Data Jadwal Mata Kuliah Tambahan  
 Sumber: Perancangan



#### 4.5.1.10 Perancangan Antarmuka Pencetakan Kartu Kehadiran Mahasiswa

Perancangan antarmuka pencetakan Kartu Kehadiran Mahasiswa (KKM) disesuaikan dengan *data flow diagram level 1* dari proses 3.2. Pencetakan KKM berfungsi untuk mencetak kartu berisi nama mahasiswa, Nomor Induk Mahasiswa (NIM), dan NIM dalam bentuk *barcode* berdasarkan tabel `m_mhs`. Masukan pencetakan digunakan untuk mengisi kata kunci yang digunakan dalam proses pencetakan KKM. Administrator dapat mengisi nim mahasiswa atau nama mahasiswa ke dalam masukan pencetakan untuk memproses pencetakan KKM. Tombol *print* digunakan untuk memulai proses pencetakan KKM. Menu pilihan pencetakan akan ditampilkan setelah administrator menekan tombol *print*. Perancangan antarmuka pencetakan KKM ditunjukkan dalam Gambar 4.60.

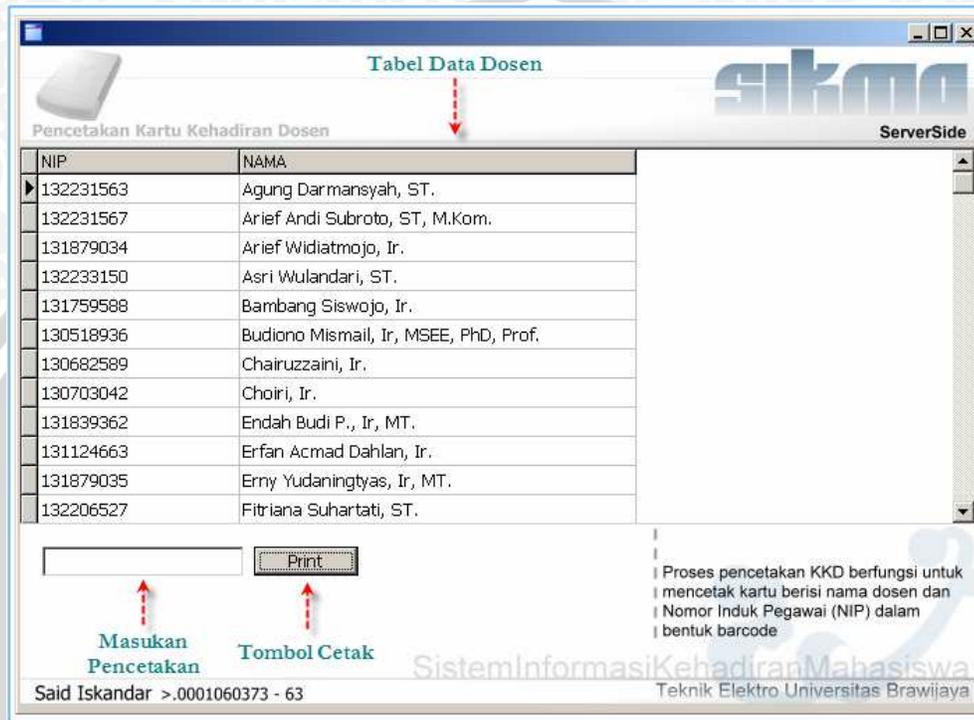


**Gambar 4.60** Antarmuka Pencetakan Kartu Kehadiran Mahasiswa  
Sumber: Perancangan

#### 4.5.1.11 Perancangan Antarmuka Pencetakan Kartu Kehadiran Dosen

Perancangan antarmuka pencetakan Kartu Kehadiran Dosen (KKD) disesuaikan dengan *data flow diagram level 1* dari proses 3.3. Pencetakan KKD berfungsi untuk mencetak kartu berisi nama dosen, Nomor Induk Dosen (NIP), dan NIP dalam bentuk *barcode* berdasarkan tabel `m_dosen`. Masukan pencetakan digunakan untuk mengisi kata kunci yang digunakan dalam proses pencetakan KKD. Administrator dapat

mengisikan NIP dosen ke dalam masukan pencetakan untuk memproses pencetakan KKD. Tombol *print* digunakan untuk memulai proses pencetakan KKD. Menu pilihan pencetakan akan ditampilkan setelah administrator menekan tombol *print*. Perancangan antarmuka pencetakan KKD ditunjukkan dalam Gambar 4.61.

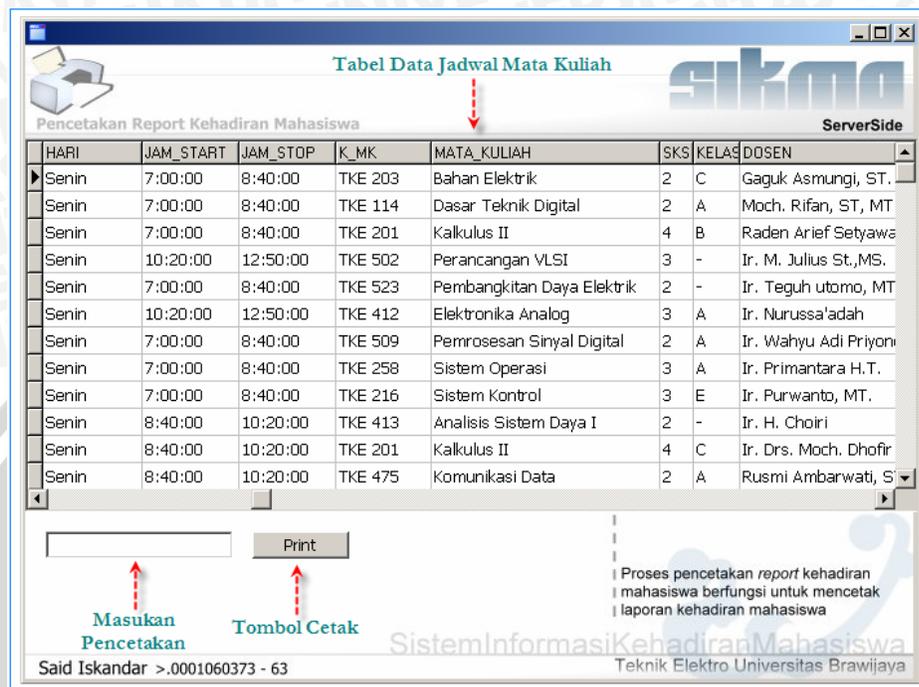


**Gambar 4.61** Antarmuka Pencetakan Kartu Kehadiran Dosen  
Sumber: *Perancangan*

#### 4.5.1.12 Perancangan Antarmuka Pencetakan *Report* Kehadiran Mahasiswa

Perancangan antarmuka pencetakan *report* kehadiran mahasiswa disesuaikan dengan *data flow diagram level 1* dari proses 3.1. Pencetakan *report* kehadiran mahasiswa berfungsi untuk mencetak laporan kehadiran mahasiswa berdasarkan tabel *d\_hadir*. Masukan pencetakan digunakan untuk mengisikan kata kunci yang digunakan dalam proses pencetakan *report* kehadiran mahasiswa. Administrator dapat mengisikan id jadwal mata kuliah ke dalam masukan pencetakan untuk memproses pencetakan *report* kehadiran mahasiswa. Tombol *print* digunakan untuk memulai proses pencetakan *report* kehadiran mahasiswa. Menu pilihan pencetakan akan ditampilkan setelah administrator menekan tombol *print*.

Perancangan antarmuka pencetakan *report* kehadiran mahasiswa ditunjukkan dalam Gambar 4.62.



Gambar 4.62 Antarmuka Pencetakan *Report* Mahasiswa  
Sumber: Perancangan

#### 4.5.2 Perancangan Antarmuka Aplikasi *Client*

Perancangan antarmuka aplikasi *client* disesuaikan dengan *data flow diagram level 0* aplikasi *client*. Perancangan aplikasi *client* berfungsi untuk proses *login client* dan pencatatan kehadiran mahasiswa. Aplikasi *client* dipilah berdasarkan pengguna, yaitu dosen dan mahasiswa. Dosen melakukan otentifikasi *login* untuk pengecekan jadwal mengajar mata kuliah dan kelas mata kuliah. Mahasiswa melakukan pencatatan kehadiran setelah melalui pengecekan daftar peserta mata kuliah.

Proses *login client* ditujukan untuk membatasi akses Sistem Informasi Kehadiran Mahasiswa dari pengguna yang tidak mempunyai hak akses terhadap aplikasi *client*. *User* yang menggunakan aplikasi *client* pada Sistem Informasi Kehadiran Mahasiswa harus melalui proses otentifikasi *login client*.

Kolom masukan *barcode* digunakan dalam proses *login client*. Dosen mengisi kolom masukan *barcode* dengan kartu kehadiran dosen yang akan dibaca *barcode scanner*. NIP dosen berbentuk *barcode* akan disesuaikan dengan NIP dosen pada tabel *m\_dosen*. Pengecekan jadwal mata kuliah dan kelas mata kuliah akan dilakukan setelah melalui proses pengecekan NIP dosen. Dosen yang mempunyai jadwal mengajar sesuai

dengan jam dan hari pada aplikasi *server* dapat mengaktifkan proses pencatatan kehadiran mahasiswa.

Menu *exit* berfungsi untuk menutup aplikasi *client*. Tanggal dan jam menunjukkan waktu saat aplikasi *client* dijalankan sesuai dengan waktu pada aplikasi *server*. Nama dosen pengajar menampilkan nama dari dosen pengajar yang aktif sesuai dengan proses *login*. Nama mata kuliah menampilkan nama dari mata kuliah yang aktif sesuai dengan proses *login* dosen pengajar. Kelas mata kuliah menampilkan kelas dari mata kuliah yang aktif sesuai dengan proses *login* dosen pengajar. Kolom *input barcode* digunakan untuk proses *login* dosen dan proses pencatatan kehadiran mahasiswa. Status *login* menampilkan keadaan aplikasi aktif pada saat dosen mengaktifkan aplikasi *client* melalui *input barcode*. Perancangan antarmuka aplikasi *client* ditunjukkan dalam Gambar 4.63.

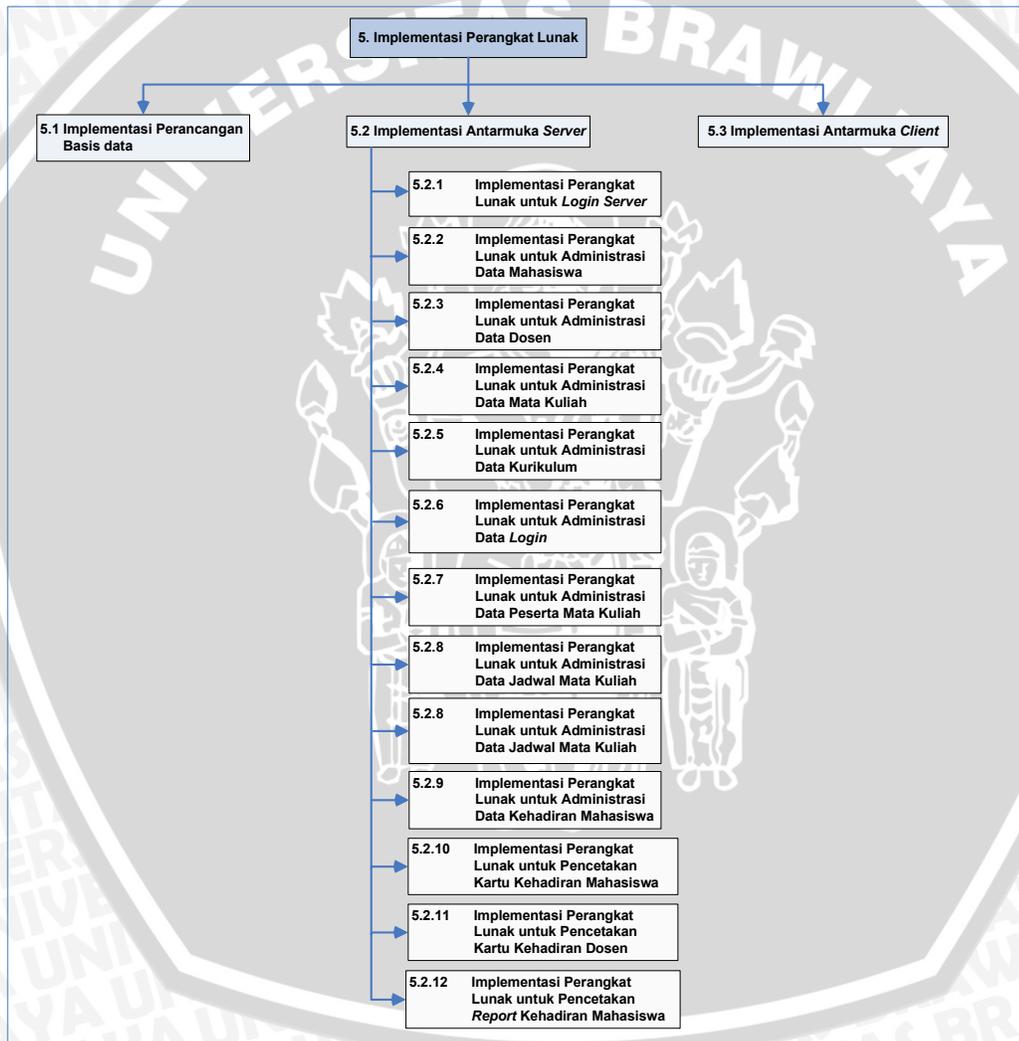


Gambar 4.63 Antarmuka Aplikasi *Client*

Sumber: Perancangan

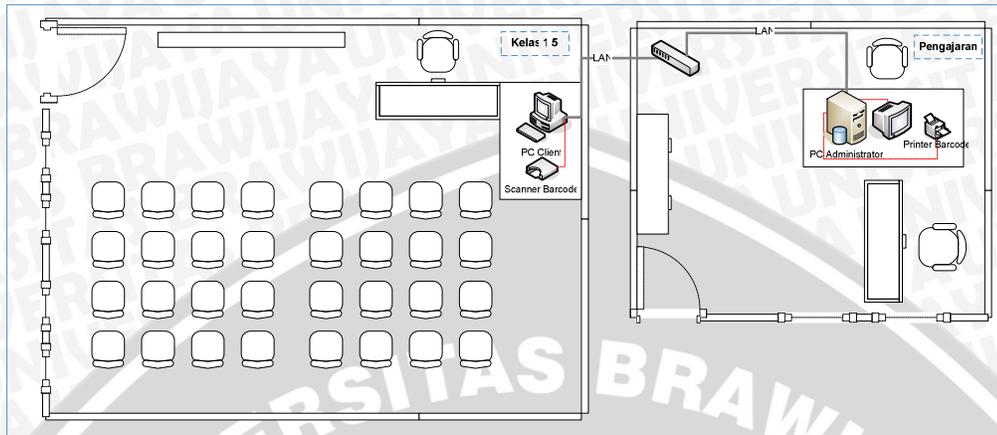
## BAB V IMPLEMENTASI PERANGKAT LUNAK

Bab ini membahas mengenai implementasi Sistem Informasi Kehadiran Mahasiswa menggunakan *barcode*. Implementasi yang dilakukan menggunakan Borland Delphi (versi 7), *server* basis data Borland Interbase (versi 6.5), dan jaringan komputer dengan menggunakan protokol TCP/IP. Diagram pohon implementasi perangkat lunak ditunjukkan dalam Gambar 5.1



**Gambar 5.1** Diagram Pohon Implementasi Perangkat Lunak  
Sumber: *Perancangan*

Implementasi Sistem Informasi Kehadiran Mahasiswa ditunjukkan dalam Gambar 5.2.



**Gambar 5.2** Implentasi Sistem Informasi Kehadiran Mahasiswa  
Sumber: Perancangan

Sistem Informasi Kehadiran Mahasiswa dibagi atas dua bagian utama, yaitu implementasi *server* pada ruang pengajaran dan *client* pada ruang kelas mata kuliah.

Ruang pengajaran terdiri dari *Personal Computer* (PC) administrator, monitor, dan *printer barcode*. Aplikasi *server* dan *Interbase 6.5 server* terletak pada PC administrator.

Aplikasi *Server* digunakan untuk administrasi data mahasiswa, dosen, mata kuliah, kurikulum, peserta mata kuliah, jadwal mata kuliah, kehadiran mahasiswa, *login*, pencetakan kartu kehadiran mahasiswa, pencetakan kartu kehadiran dosen, dan pencetakan *report* kehadiran mahasiswa.

Ruang kelas terdiri dari *Personal Computer* (PC) *client*, monitor, dan *scanner barcode*. Aplikasi *client* dan *Interbase 6.5 client* terletak pada PC *client*.

Aplikasi *SIKMA Client* digunakan untuk mengaktifkan kelas mata kuliah oleh dosen pengajar dan menyimpan data kehadiran mahasiswa melalui pembacaan kartu kehadiran mahasiswa.

Aplikasi *client* akan mengirimkan data *id* jadwal dan *nim* kepada aplikasi *server* pada saat proses otentifikasi *login* dosen dan mahasiswa. Aplikasi *client* juga mengirimkan data kehadiran mahasiswa berupa *nim* untuk disimpan di *Interbase Server* pada saat proses pencatatan kehadiran mahasiswa.

## 5.1 Implementasi Perancangan Basis Data

Implementasi perancangan basis data kehadiran dilakukan sesuai dengan *entity relationship diagram*. Implementasi perancangan basis data kehadiran menggunakan *Data Definition Language* (DDL). DDL adalah struktur basis data yang menggambarkan desain basis data secara keseluruhan [FAT-02:15].

DDL yang digunakan dalam membentuk basis data kehadiran adalah sebagai berikut :

```
create database kehadiran;
```

DDL yang digunakan dalam membentuk tabel `m_login` adalah sebagai berikut

```
create table m_login (  
k_login varchar(20) not null,  
username varchar(20) not null,  
passwd varchar(20) not null,  
hak_akses char(1),  
primary key (k_login, username));
```

DDL yang digunakan dalam membentuk tabel `m_dosen` adalah sebagai berikut

```
create table m_dosen (  
nip varchar(20) not null,  
nama varchar(40),  
alamat varchar(60),  
no_telp varchar(20),  
k_login varchar(20) not null,  
primary key (nip),  
foreign key (k_login) references m_login (k_login) on  
update cascade on delete cascade)
```

DDL yang digunakan dalam membentuk tabel `m_krk` adalah sebagai berikut :

```
create table m_krk (  
k_krk varchar(20) not null,  
nama varchar(40),  
semester char(6),  
krk_now char(1),  
primary key (k_krk));
```

DDL yang digunakan dalam membentuk tabel `m_mhs` adalah sebagai berikut :

```
create table m_mhs (  
nim varchar(20) not null,  
nama varchar(40),  
alamat varchar(60),  
primary key (nim));
```

DDL yang digunakan dalam membentuk tabel `m_mk` adalah sebagai berikut :

```
create table m_mk (  
k_mk varchar(20) not null,  
nama varchar(40),  
sks char(1),  
primary key (k_mk));
```

DDL yang digunakan dalam membentuk tabel `m_hari` adalah sebagai berikut :

```
create table m_hari (  
k_hari varchar(20) not null,  
nama varchar(40),  
primary key (k_hari));
```

DDL yang digunakan dalam membentuk tabel `d_jadwal` adalah sebagai berikut:

```
create table d_jadwal (  
id_jadwal smallint not null,  
k_krk varchar(20) not null,  
nip varchar(20) not null,  
k_mk varchar(20) not null,  
kelas_mk char(1),  
k_hari varchar(20) not null,  
jam_start time,  
jam_stop time,  
primary key (id_jadwal),  
foreign key (k_krk) references m_krk (k_krk) on update cascade  
on delete cascade,  
foreign key (nip) references m_dosen (nip) on update cascade  
on delete cascade,  
foreign key (k_mk) references m_mk (k_mk) on update cascade  
on delete cascade,  
foreign key (k_hari) references m_hari (k_hari) on update  
cascade on delete cascade);
```

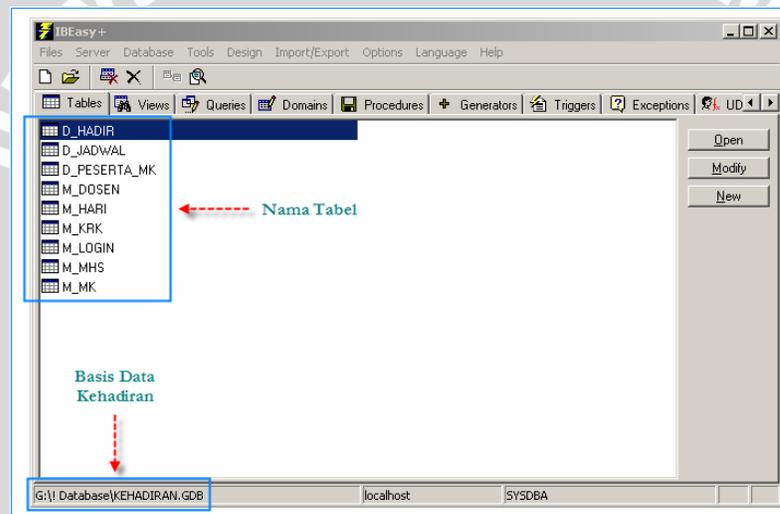
DDL yang digunakan dalam membentuk tabel `d_peserta_mk` adalah sebagai berikut :

```
create table d_peserta_mk (  
id_peserta_mk smallint not null,  
id_jadwal smallint not null,  
nim varchar(20) not null,  
primary key (id_peserta_mk),  
foreign key (id_jadwal) references d_jadwal (id_jadwal) on  
update cascade on delete cascade),  
foreign key (nim) references m_mhs (nim) on update cascade  
on delete cascade);
```

DDL yang digunakan dalam membentuk tabel `d_hadir` adalah sebagai berikut

```
create table d_hadir (
  id_hadir smallint not null,
  id_jadwal smallint not null,
  nim varchar(20) not null,
  tanggal date,
  jam time,
  primary key (id_hadir),
  foreign key (id_jadwal) references d_jadwal (id_jadwal) on
  update cascade on delete cascade,
  foreign key (nim) references m_mhs (nim) on update cascade
  on delete cascade);
```

Basis data kehadiran yang diimplementasikan pada basis data Borland Interbase ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Implementasi Basis Data kehadiran pada Borland Interbase  
Sumber: *Implementasi*

## 5.2 Implementasi Antarmuka Aplikasi Server

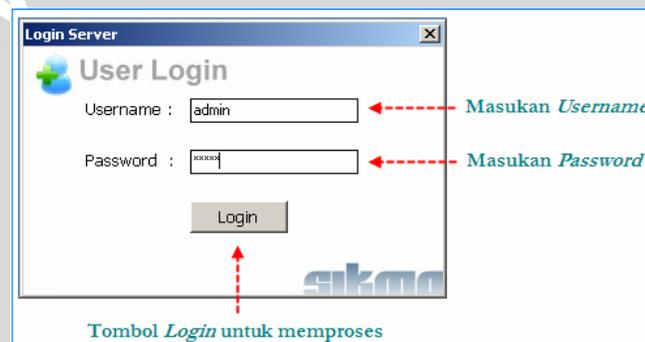
Implementasi antarmuka aplikasi *server* terdiri dari:

1. Implementasi perangkat lunak untuk *login server*.
2. Implementasi perangkat lunak untuk administrasi data mahasiswa.
3. Implementasi perangkat lunak untuk administrasi data dosen.
4. Implementasi perangkat lunak untuk administrasi data mata kuliah.
5. Implementasi perangkat lunak untuk administrasi data kurikulum.
6. Implementasi perangkat lunak untuk administrasi data *login*.
7. Implementasi perangkat lunak untuk administrasi data peserta mata kuliah.
8. Implementasi perangkat lunak untuk administrasi jadwal mata kuliah.

9. Implementasi perangkat lunak untuk administrasi data kehadiran mahasiswa.
10. Implementasi perangkat lunak untuk pencetakan kartu kehadiran mahasiswa.
11. Implementasi perangkat lunak untuk pencetakan kartu kehadiran dosen.
12. Implementasi perangkat lunak untuk pencetakan *report* kehadiran mahasiswa.

### 5.2.1 Implementasi Perangkat Lunak untuk *Login Server*

Administrator yang menggunakan aplikasi *server* pada Sistem Informasi Kehadiran Mahasiswa harus melalui proses otentifikasi *login server*. Proses otentifikasi *login server* merupakan proses pengecekan masukan *username* dan *password*, dan hak akses dibandingkan dengan yang terdapat pada tabel *m\_login* di basis data kehadiran. Implementasi antarmuka *login server* ditunjukkan dalam Gambar 5.4.



**Gambar 5.4** Implementasi Antarmuka *Login Server*  
Sumber: *Implementasi*

Pesan peringatan akan ditampilkan bila pengecekan *username* dan *password* tidak sesuai dengan yang terdapat pada tabel *m\_login* dan pengguna tidak mempunyai hak akses sebagai administrator. Aplikasi *server* akan ditutup jika terdapat kesalahan dalam memasukkan *login* lebih dari tiga kali. Proses *login* dapat dilakukan kembali dengan membuka aplikasi *server* setelah melakukan kesalahan *login* sebanyak tiga kali. Implementasi pesan peringatan ditunjukkan dalam Gambar 5.5.



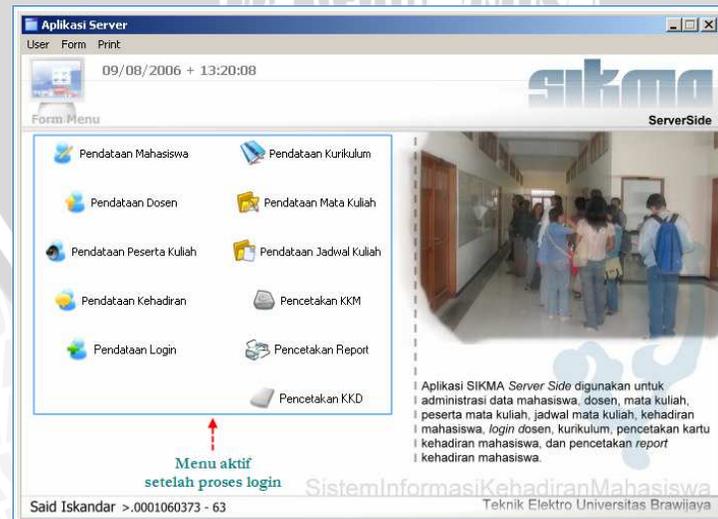
**Gambar 5.5** Implementasi Pesan Peringatan *Login Server*  
Sumber: *Implementasi*

Menu pada aplikasi *server* tidak aktif sebelum administrator melakukan proses *login server*. Proses *login server* menjaga dari pengguna yang tidak mempunyai hak terhadap aplikasi *server*. Implementasi menu tidak aktif pada aplikasi *server* sebelum proses *login* ditunjukkan dalam Gambar 5.6.



**Gambar 5.6** Implementasi Menu Tidak Aktif sebelum Proses Login  
Sumber: Implementasi

Proses *login server* akan mengaktifkan menu pada *aplikasi server* bila pengecekan *username* dan *password* sesuai dengan yang terdapat pada tabel *m\_login* dan pengguna mempunyai hak akses sebagai administrator. Implementasi menu aktif pada aplikasi *server* setelah proses *login* ditunjukkan dalam Gambar 5.7.



**Gambar 5.7** Implementasi Menu Aktif setelah Proses Login  
Sumber: Implementasi

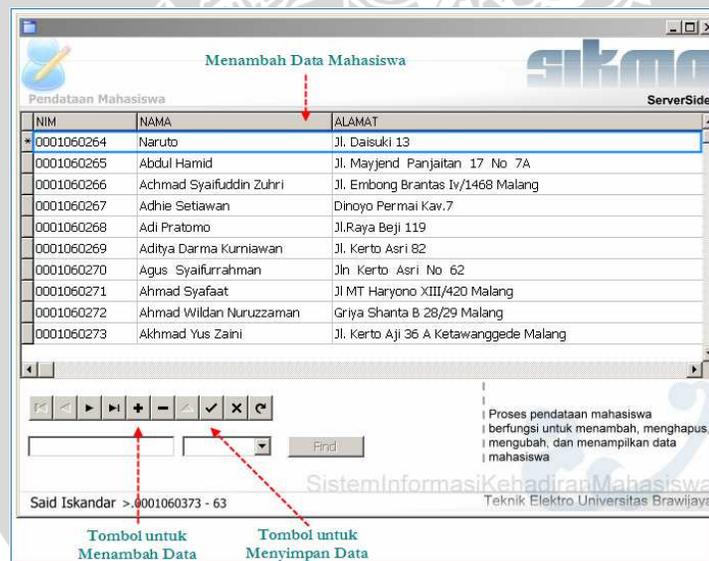
## 5.2.2 Implementasi Perangkat Lunak untuk Administrasi Data Mahasiswa

Administrasi data mahasiswa merupakan proses manajemen data mahasiswa pada tabel `m_mhs` di basis data kehadiran. Manajemen data mahasiswa meliputi menambah data, mengubah data, menghapus data, dan mencari data mahasiswa pada tabel data mahasiswa.

### 5.2.2.1 Implementasi Penambahan Data Mahasiswa

Proses menambah data mahasiswa dilakukan dengan memasukkan data mahasiswa baru ke dalam tabel data mahasiswa. Data mahasiswa baru yang dimasukkan akan disimpan dalam tabel `m_mhs`.

Administrator dapat menambah data mahasiswa dengan menekan tombol navigasi bertanda (+) dan memasukkan data mahasiswa baru ke dalam tabel data mahasiswa. Administrator dapat menyimpan data mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓). Implementasi penambahan data mahasiswa ditunjukkan dalam Gambar 5.8.



**Gambar 5.8** Implementasi Penambahan Data Mahasiswa

Sumber: *Implementasi*

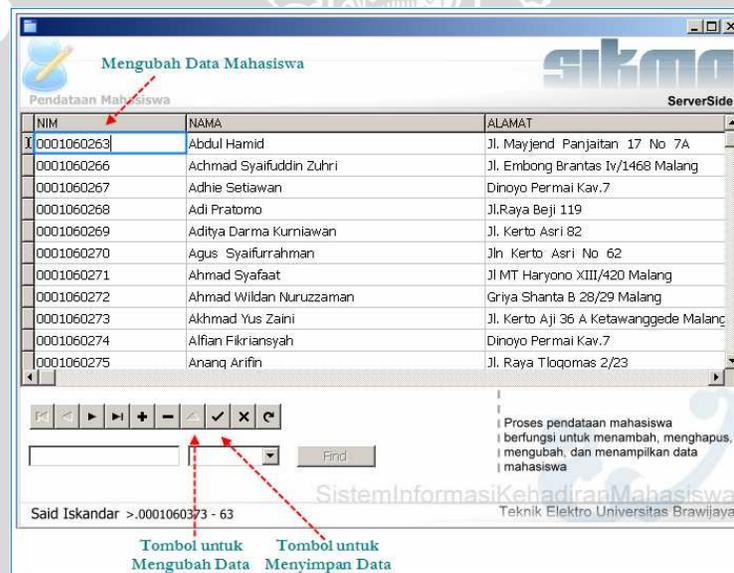
Administrator menempatkan posisi kursor pada *record* pertama dengan NIM '0001060265' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data mahasiswa baru. Administrator mengisi *record* kosong dengan NIM '0001060264', NAMA 'Naruto', dan ALAMAT 'Jl.Daisuki 13'.

Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_mhs`.

### 5.2.2.2 Implementasi Pengubahan Data Mahasiswa

Proses mengubah data mahasiswa dilakukan dengan mengubah data mahasiswa yang terdapat dalam tabel data mahasiswa. Data mahasiswa yang diubah akan disimpan dalam tabel `m_mhs`.

Administrator dapat mengubah data mahasiswa dengan menekan tombol navigasi bertanda (▲) dan memasukkan data mahasiswa yang akan diubah ke dalam tabel data mahasiswa. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data sedang dilakukan. Administrator dapat menyimpan data mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi pengubahan data mahasiswa ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Implementasi Pengubahan Data Mahasiswa

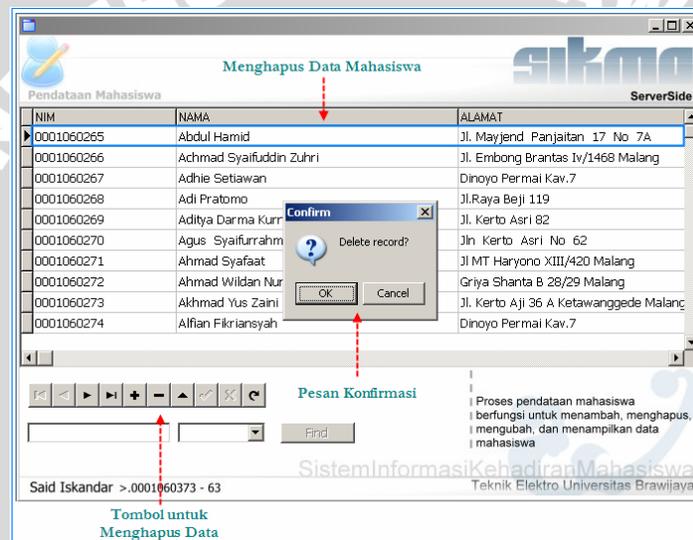
Sumber: *Implementasi*

Administrator menempatkan posisi kursor pada *record* pertama dengan NIM ‘0001060265’ dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data. Administrator mengganti NIM ‘0001060265’ dengan NIM ‘0001060263’. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel data `m_mhs`.

### 5.2.2.3 Implementasi Penghapusan Data Mahasiswa

Proses menghapus data mahasiswa dilakukan dengan menghapus data mahasiswa terdapat dalam tabel data mahasiswa. Data mahasiswa yang dihapus akan dihilangkan dari tabel `m_mhs`.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data mahasiswa akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data mahasiswa dengan menekan tombol navigasi bertanda (–). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang dihapus. Implementasi penghapusan data mahasiswa ditunjukkan dalam Gambar 5.10.



Gambar 5.10 Implementasi Penghapusan Data Mahasiswa

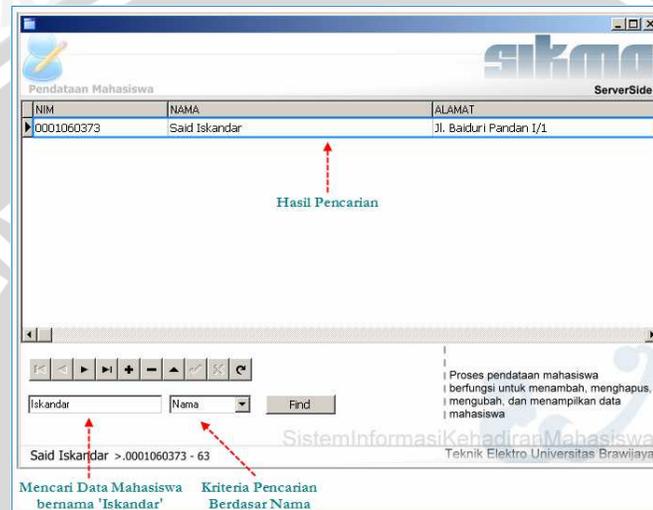
Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan NIM '0001060265', NAMA 'Abdul Hamid', dan ALAMAT 'Jl. Mayjend Panjaitan 17 No. 7A' dengan menekan tombol navigasi bertanda (–) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `m_mhs`.

### 5.2.2.4 Implementasi Pencarian Data Mahasiswa

Proses pencarian data mahasiswa merupakan proses untuk mencari data mahasiswa sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan

pencarian. Administrator menekan tombol cari atau tombol `enter` untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data mahasiswa bila ditemukan kesesuaian antara kata kunci pencarian dengan data mahasiswa pada tabel `m_mhs`. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian. Implementasi pencarian data mahasiswa ditunjukkan dalam Gambar 5.11.



**Gambar 5.11** Implementasi Pencarian Data Mahasiswa  
Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa Nama dan mengisi kata kunci pencarian dengan Nama 'Iskandar' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol `enter` untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan Nama 'Iskandar' menghasilkan keluaran pencarian pada tabel data mahasiswa berupa *record* data mahasiswa dengan NIM '0001060373', NAMA 'Said Iskandar', dan ALAMAT 'Jl. Baiduri Pandan I/1'.

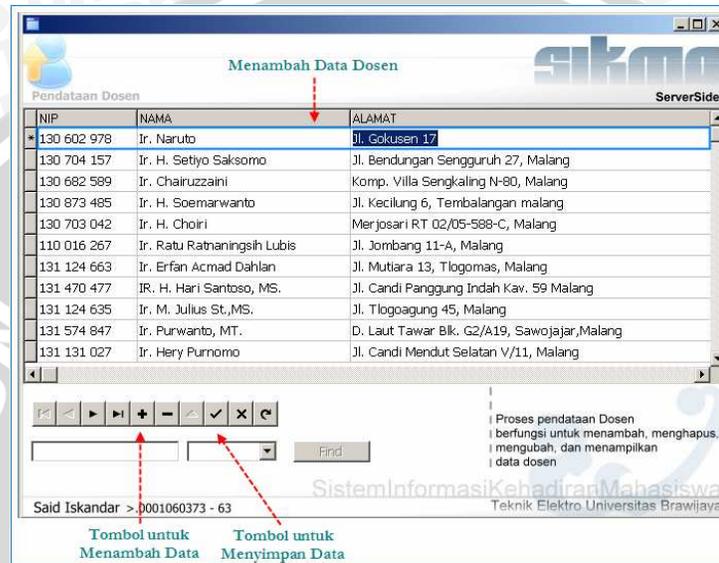
### 5.2.3 Implementasi Perangkat Lunak untuk Administrasi Data Dosen

Administrasi data dosen merupakan proses manajemen data dosen pada tabel `m_dosen` di basis data kehadiran. Manajemen data dosen meliputi menambah data, mengubah data, menghapus data, dan mencari data dosen pada tabel data dosen.

#### 5.2.3.1 Implementasi Penambahan Data Dosen

Proses menambah data dosen dilakukan dengan memasukkan data dosen baru ke dalam tabel data dosen. Data dosen baru yang dimasukkan akan disimpan dalam tabel `m_dosen`.

Administrator dapat menambah data dosen dengan menekan tombol navigasi bertanda (+) dan memasukkan data dosen baru ke dalam tabel data dosen. Administrator dapat menyimpan data dosen baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi penambahan data dosen ditunjukkan dalam Gambar 5.12.



Gambar 5.12 Implementasi Penambahan Data Dosen  
Sumber: *Implementasi*

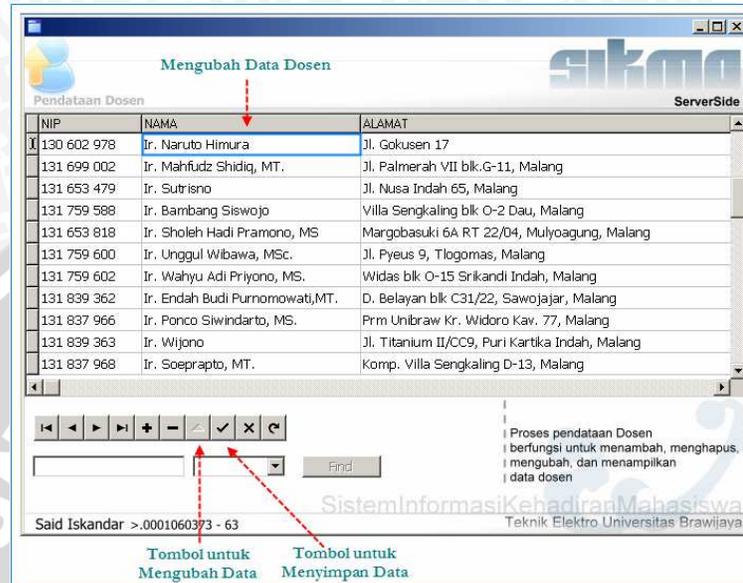
Administrator menempatkan posisi kursor pada *record* pertama dengan NIP '130 704 157' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data dosen baru. Administrator mengisi *record* kosong dengan NIP '130 602 978', NAMA 'Ir. Naruto', dan ALAMAT 'Jl. Gokusen 17'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_dosen`.

### 5.2.3.2 Implementasi Pengubahan Data Dosen

Proses mengubah data dosen dilakukan dengan mengubah data dosen yang terdapat dalam tabel data dosen. Data dosen yang diubah akan disimpan dalam tabel `m_dosen`.

Administrator dapat mengubah data dosen dengan menekan tombol navigasi bertanda (▲) dan memasukkan data dosen yang akan diubah ke dalam tabel data dosen. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data

sedang dilakukan. Administrator dapat menyimpan data dosen baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi perubahan data dosen ditunjukkan dalam Gambar 5.13.



Gambar 5.13 Implementasi Perubahan Data Dosen  
Sumber: Implementasi

Administrator menempatkan posisi kursor pada *record* pertama dengan NIP '130 602 978' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data. Administrator mengganti NAMA 'Ir. Naruto' dengan NAMA 'Ir. Naruto Himura'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_dosen`.

### 5.2.3.3 Implementasi Penghapusan Data Dosen

Proses menghapus data dosen dilakukan dengan menghapus data dosen terdapat dalam tabel data dosen. Data dosen yang dihapus akan dihilangkan dari tabel `m_dosen`.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data dosen dengan menekan tombol navigasi bertanda (–). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus.

Implementasi penghapusan data dosen ditunjukkan dalam Gambar 5.14.



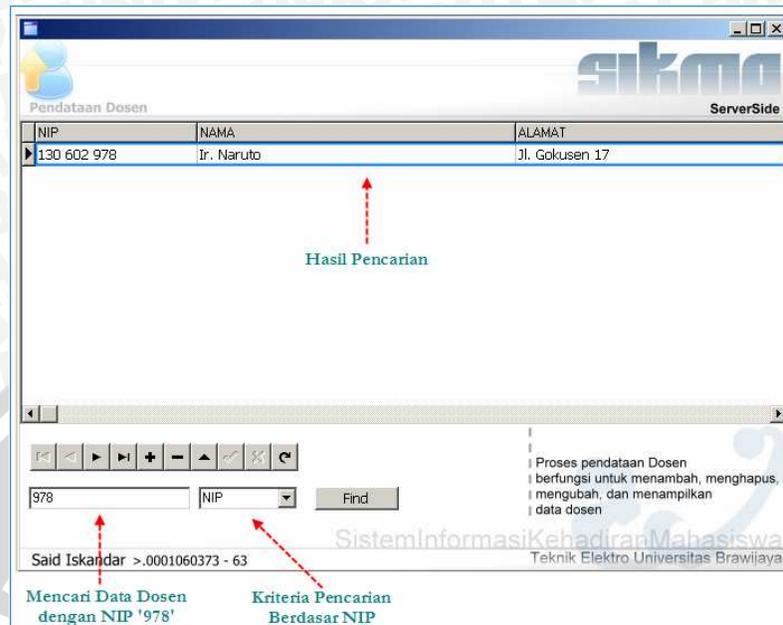
**Gambar 5.14** Implementasi Penghapusan Data Dosen  
Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan NIP ‘130 602 978’, NAMA ‘Ir. Naruto Himura’, dan ALAMAT ‘Jl. Gokusen 17’ dengan menekan tombol navigasi bertanda (-) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `m_dosen`.

#### 5.2.3.4 Implementasi Pencarian Data Dosen

Proses pencarian data dosen merupakan proses untuk mencari data dosen sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian. Administrator menekan tombol cari atau tombol enter untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data dosen bila ditemukan kesesuaian antara kata kunci pencarian dengan data dosen pada tabel `m_dosen`. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian.

Implementasi pencarian data dosen ditunjukkan dalam Gambar 5.15.



Gambar 5.15 Implementasi Pencarian Data Dosen

Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa NIP dan mengisi kata kunci pencarian dengan NIP '978' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan NIP '978' menghasilkan keluaran pencarian pada tabel data dosen berupa *record* data dosen dengan NIP '130 602 978', NAMA 'Ir. Naruto', dan ALAMAT 'Jl. Gokusen 17'.

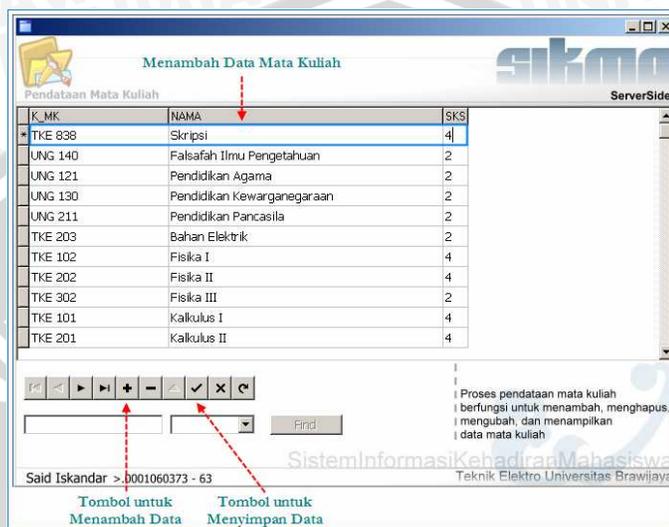
## 5.2.4 Implementasi Perangkat Lunak untuk Administrasi Data Mata Kuliah

Administrasi data mata kuliah merupakan proses manajemen data mata kuliah pada tabel `m_mk` di basis data kehadiran. Manajemen data mata kuliah meliputi menambah data, mengubah data, menghapus data, dan mencari data mata kuliah pada tabel data mata kuliah.

### 5.2.4.1 Implementasi Penambahan Data Mata Kuliah

Proses menambah data mata kuliah dilakukan dengan memasukkan data mata kuliah baru ke dalam tabel data mata kuliah. Data mata kuliah baru yang dimasukkan akan disimpan dalam tabel `m_mk`.

Administrator dapat menambah data mata kuliah dengan menekan tombol navigasi bertanda (+) dan memasukkan data mata kuliah baru ke dalam tabel data mata kuliah. Administrator dapat menyimpan data mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi penambahan data mata kuliah ditunjukkan dalam Gambar 5.16.



**Gambar 5.16** Implementasi Penambahan Data Mata Kuliah

Sumber: *Implementasi*

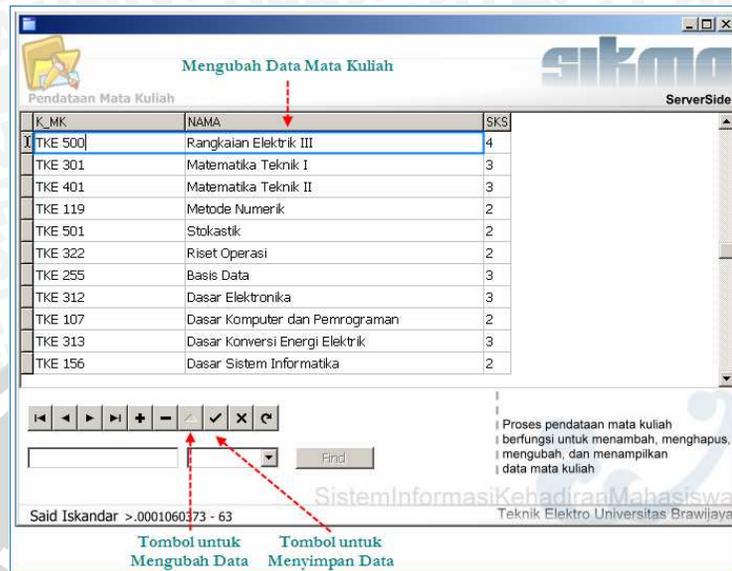
Administrator menempatkan posisi kursor pada *record* pertama dengan NAMA ‘Falsafah Ilmu Pengetahuan’ dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data mata kuliah baru. Administrator mengisi *record* kosong dengan K\_MK ‘TKE 838’, NAMA ‘Skripsi’, dan SKS ‘4’. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel m\_mk.

#### 5.2.4.2 Implementasi Pengubahan Data Mata Kuliah

Proses mengubah data mata kuliah dilakukan dengan mengubah data mata kuliah yang terdapat dalam tabel data mata kuliah. Data mata kuliah yang diubah akan disimpan dalam tabel m\_mk.

Administrator dapat mengubah data mata kuliah dengan menekan tombol navigasi bertanda (▲) dan memasukkan data mata kuliah yang akan diubah ke dalam tabel data mata kuliah. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data sedang dilakukan. Administrator dapat menyimpan data mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

Implementasi perubahan data mata kuliah ditunjukkan dalam Gambar 5.17.



**Gambar 5.17** Implementasi Perubahan Data Mata Kuliah  
Sumber: *Implementasi*

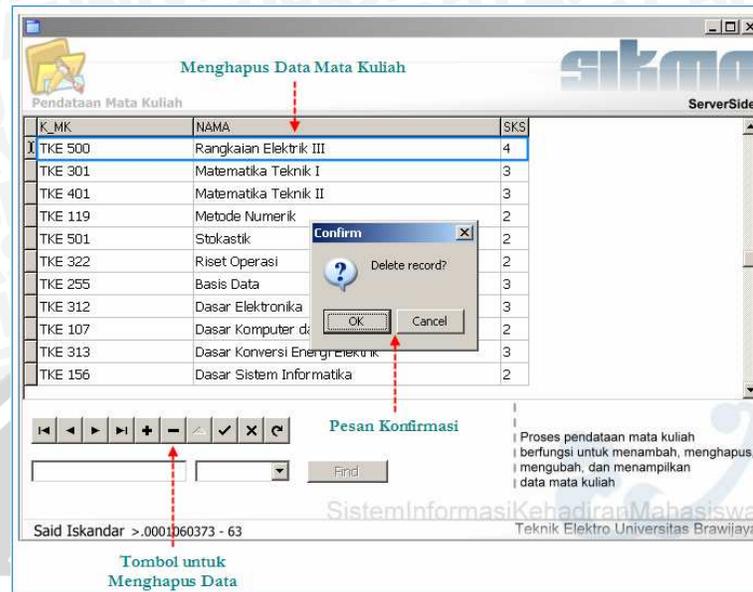
Administrator menempatkan posisi kursor pada *record* pertama dengan K\_MK 'TKE 301' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data mata kuliah. Administrator mengganti K\_MK 'TKE 260' dengan K\_MK 'TKE 500' dan NAMA 'Jaringan Komputer' dengan NAMA 'Rangkaian Elektrik III'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel m\_mk.

#### 5.2.4.3 Implementasi Penghapusan Data Mata Kuliah

Proses menghapus data mata kuliah dilakukan dengan menghapus data mata kuliah terdapat dalam tabel data mata kuliah. Data mata kuliah yang dihapus akan dihilangkan dari tabel m\_mk.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data mata kuliah dengan menekan tombol navigasi bertanda (-). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus.

Implementasi penghapusan data mata kuliah ditunjukkan dalam Gambar 5.18.



**Gambar 5.18** Implementasi Penghapusan Data Mata Kuliah

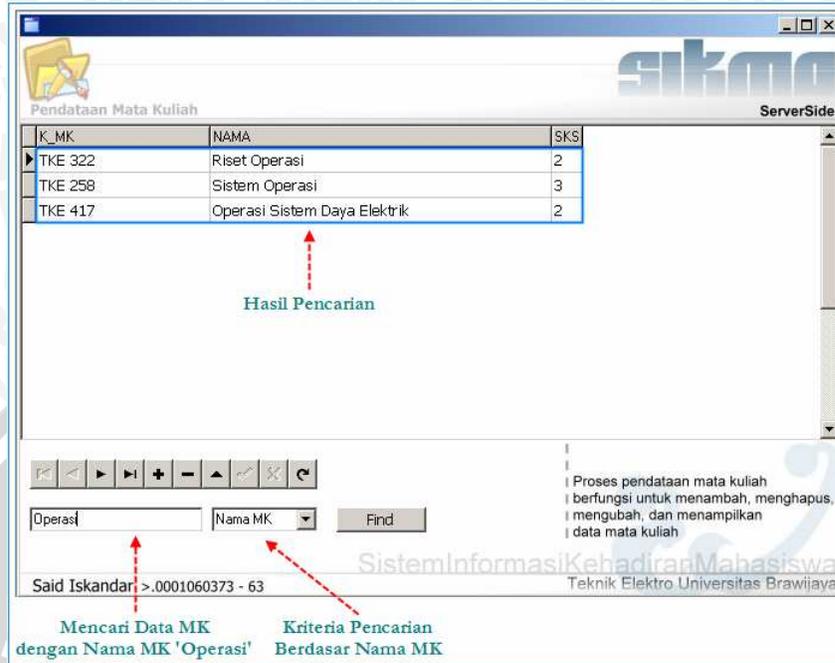
Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan K\_MK 'TKE 500', NAMA 'Rangkaian Elektrik III', dan SKS '4' dengan menekan tombol navigasi bertanda (-) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel m\_mk.

#### 5.2.4.4 Implementasi Pencarian Data Mata Kuliah

Proses pencarian data mata kuliah merupakan proses untuk mencari data mata kuliah sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian. Administrator menekan tombol cari atau tombol enter untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data mata kuliah bila ditemukan kesesuaian antara kata kunci pencarian dengan data mata kuliah pada tabel m\_mk. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian.

Implementasi pencarian data mata kuliah ditunjukkan dalam Gambar 5.19.



**Gambar 5.19** Implementasi Pencarian Data Mata Kuliah  
 Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa NAMA dan mengisi kata kunci pencarian dengan NAMA ‘Operasi’ pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan NAMA ‘Operasi’ menghasilkan data tiga mata kuliah yang mengandung kata ‘Operasi’. Keluaran pencarian yang ditunjukkan pada Tabel 5.1.

**Tabel 5.1** Hasil Pencarian Data Mata Kuliah

K_MK	NAMA	SKS
TKE 322	Riset Operasi	2
TKE 258	Sistem Operasi	3
TKE 417	Operasi Sistem Daya Elektrik	2

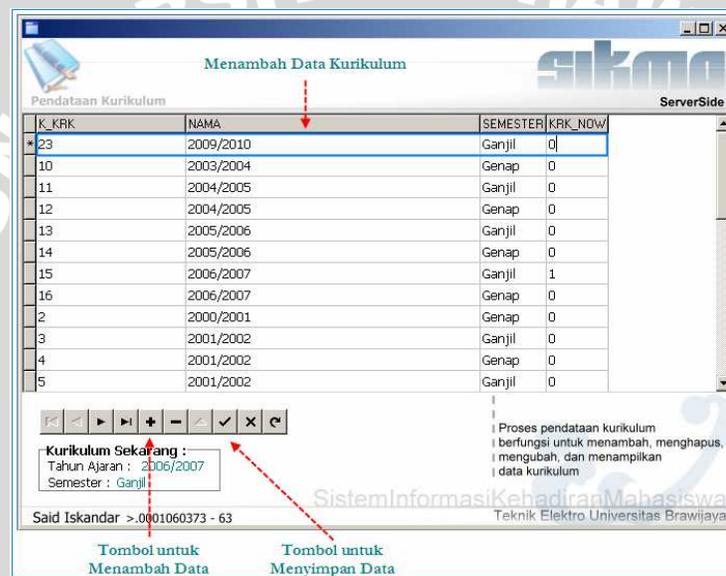
### 5.2.5 Implementasi Perangkat Lunak untuk Administrasi Data Kurikulum

Administrasi data kurikulum merupakan proses manajemen data kurikulum pada tabel `m_krk` di basis data kehadiran. Manajemen data kurikulum meliputi menambah data, mengubah data, menghapus data, dan mencari data kurikulum pada tabel `m_krk`.

### 5.2.5.1 Implementasi Penambahan Data Kurikulum

Proses menambah data kurikulum dilakukan dengan memasukkan data kurikulum baru ke dalam tabel data kurikulum. Data kurikulum baru yang dimasukkan akan disimpan dalam tabel `m_krk`.

Administrator dapat menambah data kurikulum dengan menekan tombol navigasi bertanda (+) dan memasukkan data kurikulum baru ke dalam tabel data kurikulum. Administrator dapat menyimpan data kurikulum baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi penambahan data kurikulum ditunjukkan dalam Gambar 5.20.



Gambar 5.20 Implementasi Penambahan Data Kurikulum

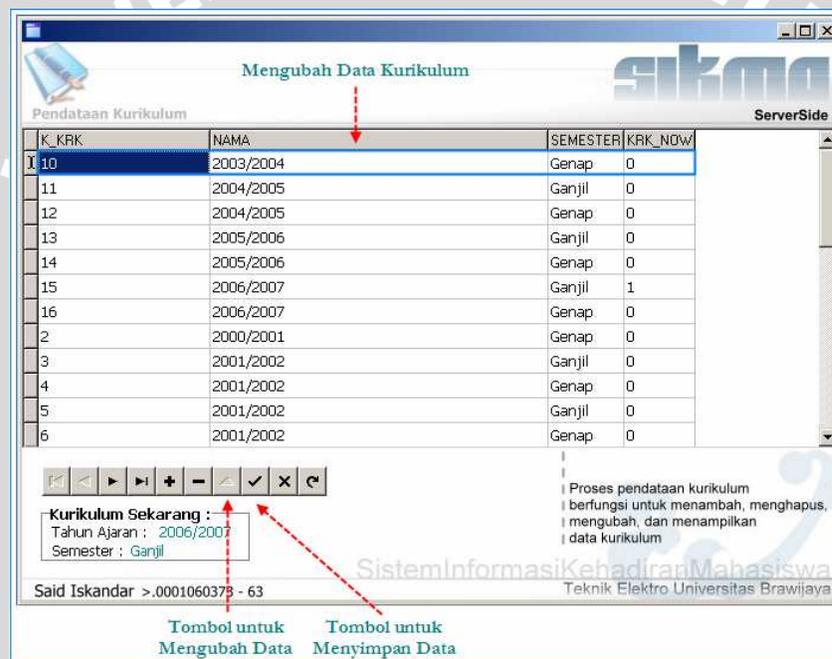
Sumber: *Implementasi*

Administrator menempatkan posisi kursor pada *record* pertama dengan K\_KRK '23' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data kurikulum baru. Administrator mengisi *record* kosong dengan K\_KRK '23', NAMA '2009/2010', SEMESTER 'Ganjil' dan KRK\_NOW '0'. KRK\_NOW '0' menunjukkan kurikulum yang tidak aktif pada tahun ajaran 2009/2010 dan semester ganjil. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_krk`.

### 5.2.5.2 Implementasi Pengubahan Data Kurikulum

Proses mengubah data kurikulum dilakukan dengan mengubah data kurikulum yang terdapat dalam tabel data kurikulum. Data kurikulum yang diubah akan disimpan dalam tabel `m_krk`.

Administrator dapat mengubah data kurikulum dengan menekan tombol navigasi bertanda (▲) dan memasukkan data kurikulum yang akan diubah ke dalam tabel data kurikulum. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data sedang dilakukan. Administrator dapat menyimpan data kurikulum baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi pengubahan data kurikulum ditunjukkan dalam Gambar 5.21.



**Gambar 5.21** Implementasi Pengubahan Data Kurikulum

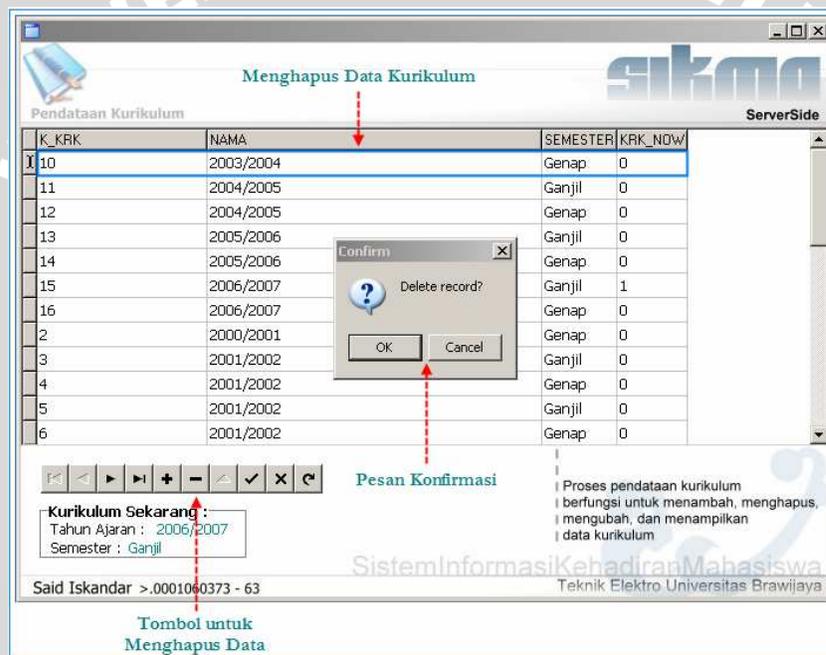
Sumber: *Implementasi*

Administrator menempatkan posisi kursor pada *record* pertama dengan `K_KRK` '10' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data kurikulum. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_krk`.

### 5.2.5.3 Implementasi Penghapusan Data Kurikulum

Proses menghapus data kurikulum dilakukan dengan menghapus data kurikulum terdapat dalam tabel data kurikulum. Data kurikulum yang dihapus akan dihilangkan dari tabel `m_krk`.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data kurikulum dengan menekan tombol navigasi bertanda (–). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus. Implementasi penghapusan data kurikulum ditunjukkan dalam Gambar 5.22.



**Gambar 5.22** Implementasi Penghapusan Data Kurikulum

Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan `K_KRK` '10', `NAMA` '2003/2004', `SEMESTER` 'Genap', dan `KRK_NOW` '0' dengan menekan tombol navigasi bertanda (–) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `m_krk`.

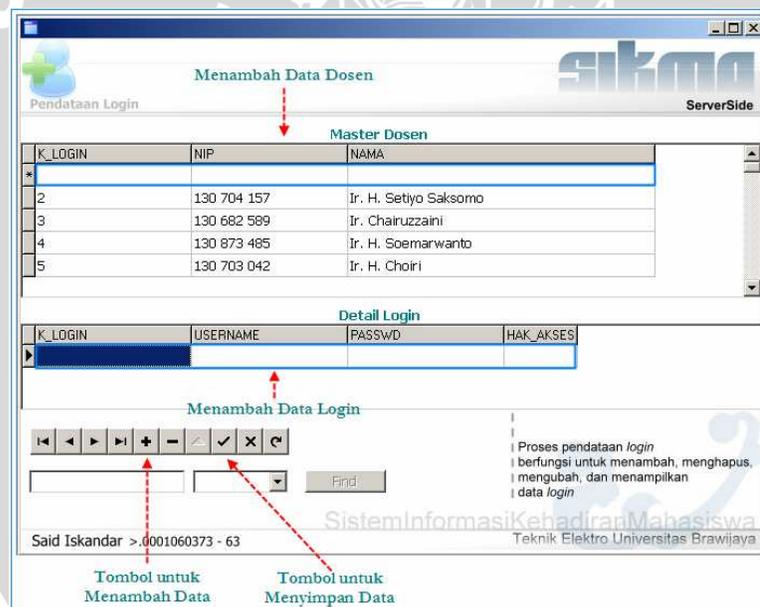
## 5.2.6 Implementasi Perangkat Lunak untuk Administrasi Data Login

Administrasi data *login* merupakan proses manajemen data *login* pada tabel *m\_login* di basis data kehadiran. Manajemen data *login* meliputi menambah data, mengubah data, menghapus data, dan mencari data *login* pada tabel *m\_login*.

### 5.2.6.1 Implementasi Penambahan Data Login

Proses menambah data *login* dilakukan dengan memasukkan data *login* baru ke dalam tabel data *login*. Data *login* baru yang dimasukkan akan disimpan dalam tabel *m\_login*.

Administrator dapat menambah data *login* dengan menekan tombol navigasi bertanda (+) dan memasukkan data *login* baru ke dalam tabel data *login*. Administrator dapat menyimpan data *login* baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓). Implementasi penambahan data *login* ditunjukkan dalam Gambar 5.23.



Gambar 5.23 Implementasi Penambahan Data Login  
Sumber: Implementasi

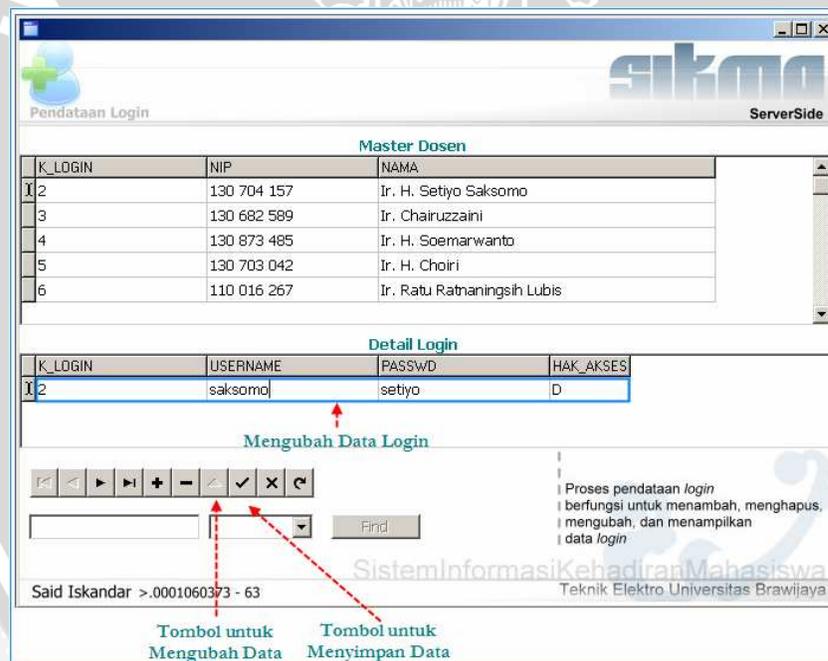
Administrator dapat mengisi data *login* dosen setelah menambahkan data dosen baru. Administrator menempatkan posisi kursor pada *record* pertama dengan NIP '130 704 157' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data dosen baru. Administrator dapat menambahkan data detail *login* dosen sesuai dengan yang diisi sebelumnya pada tabel master dosen.

Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_login`.

### 5.2.6.2 Implementasi Pengubahan Data *Login*

Proses mengubah data *login* dilakukan dengan mengubah data *login* yang terdapat dalam tabel data *login*. Data *login* yang diubah akan disimpan dalam tabel `m_login`.

Administrator dapat mengubah data *login* dengan menekan tombol navigasi bertanda (▲) dan memasukkan data *login* yang akan diubah ke dalam tabel data *login*. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data sedang dilakukan. Administrator dapat menyimpan data *login* baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi pengubahan data *login* ditunjukkan dalam Gambar 5.24.



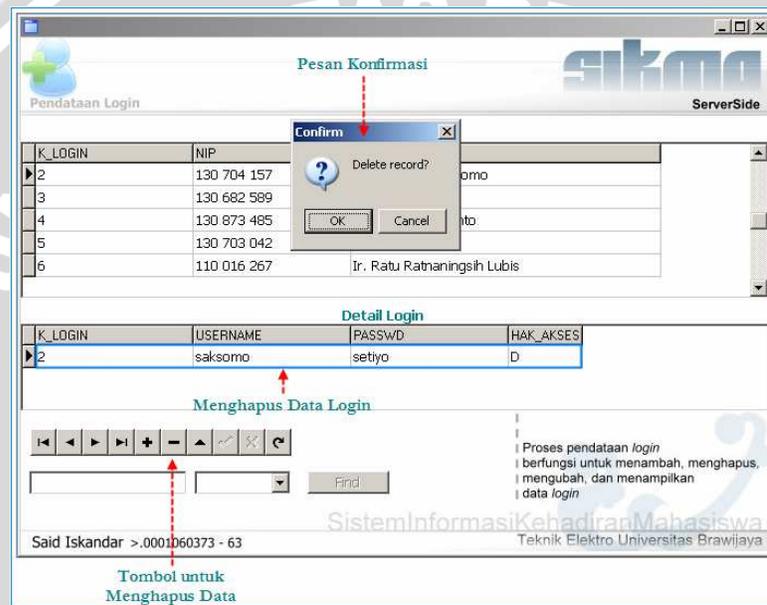
**Gambar 5.24** Implementasi Pengubahan Data *Login*  
Sumber: *Implementasi*

Administrator menempatkan posisi kursor pada *record* pertama dengan `K_LOGIN` '2' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data *login*. Administrator mengganti USERNAME 'setiyo' dengan USERNAME 'saksomo'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `m_login`.

### 5.2.6.3 Implementasi Penghapusan Data Login

Proses menghapus data *login* dilakukan dengan menghapus data *login* terdapat dalam tabel data *login*. Data *login* yang dihapus akan dihilangkan dari tabel *m\_login*.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data *login* dengan menekan tombol navigasi bertanda (←). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus. Implementasi penghapusan data *login* ditunjukkan dalam Gambar 5.25.



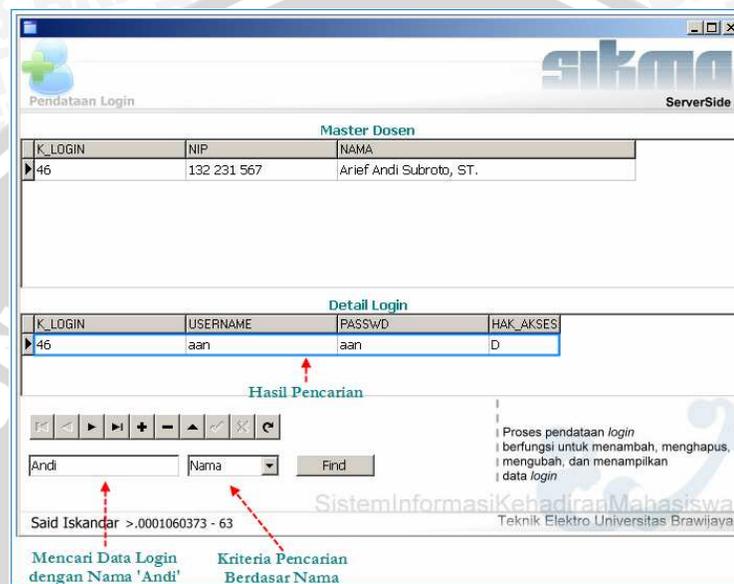
Gambar 5.25 Implementasi Penghapusan Data Login  
Sumber: Implementasi

Administrator akan menghapus *record* data dengan *K\_LOGIN* '2', *USERNAME* 'saksomo', *PASSWD* 'setiyo', dan *HAK\_AKSES* 'D' dengan menekan tombol navigasi bertanda (←) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel *m\_login*.

### 5.2.6.4 Implementasi Pencarian Data Login

Proses pencarian data *login* merupakan proses untuk mencari data *login* sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.

Administrator menekan tombol cari atau tombol `enter` untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data *login* bila ditemukan kesesuaian antara kata kunci pencarian dengan data *login* pada tabel *m\_login*. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian. Implementasi pencarian data *login* ditunjukkan dalam Gambar 5.26.



**Gambar 5.26** Implementasi Pencarian Data *Login*  
Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa NAMA dan mengisi kata kunci pencarian dengan NAMA 'Andi' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol `enter` untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan NAMA 'Andi' menghasilkan keluaran pencarian pada tabel data *login* berupa *record* data dosen dengan K\_LOGIN '46', NIP '132 231 567', NAMA 'Arief Andi Subroto, ST., M.Kom.', dan *record* data *login* K\_LOGIN '46', USERNAME 'aan', PASSWD 'aan', HAK\_AKSES 'D'.

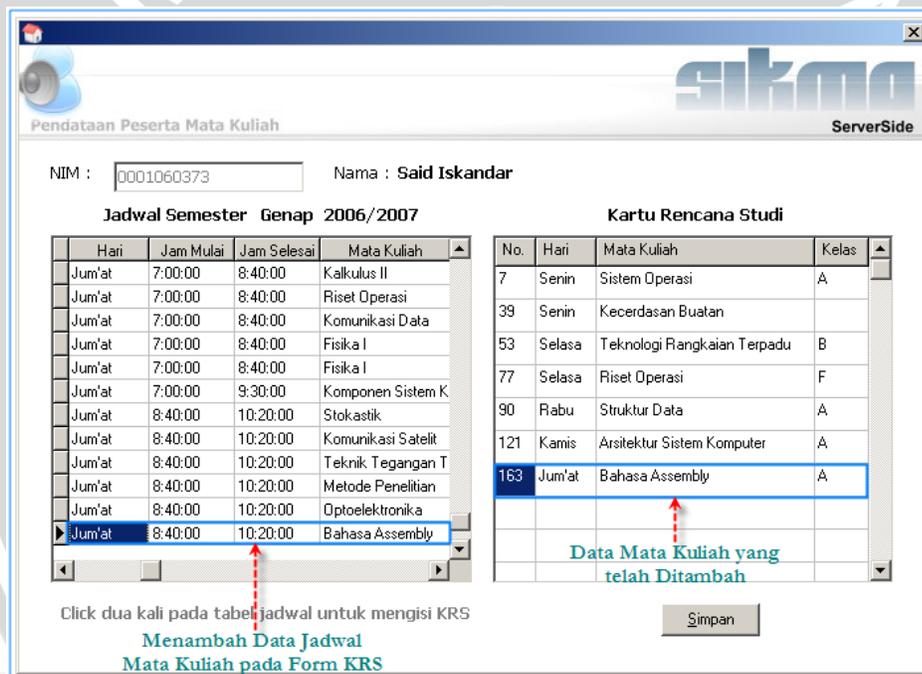
### 5.2.7 Implementasi Perangkat Lunak untuk Administrasi Data Peserta Mata Kuliah

Administrasi data peserta mata kuliah merupakan proses manajemen data peserta mata kuliah pada tabel *d\_peserta\_mk* di basis data kehadiran. Manajemen data peserta mata kuliah meliputi menambah data, mengubah data, dan menghapus data peserta mata kuliah pada tabel *d\_peserta\_mk*.

### 5.2.7.1 Implementasi Penambahan Data Peserta Mata Kuliah

Proses menambah data peserta mata kuliah dilakukan dengan memasukkan data peserta mata kuliah baru ke dalam tabel data peserta mata kuliah. Data peserta mata kuliah baru yang dimasukkan akan disimpan dalam tabel `d_peserta_mk`. Data peserta mata kuliah hanya dapat ditambah oleh administrator.

Administrator dapat menambah data peserta mata kuliah dengan menekan tombol kiri pada *mouse* sebanyak dua kali pada tabel jadwal mata kuliah dan memasukkan data peserta mata kuliah baru ke dalam tabel data peserta mata kuliah. Administrator dapat menyimpan data peserta mata kuliah baru pada basis data kehadiran dengan menekan tombol *Simpan*. Implementasi penambahan data peserta mata kuliah ditunjukkan dalam Gambar 5.27.



Gambar 5.27 Implementasi Penambahan Data Peserta Mata Kuliah

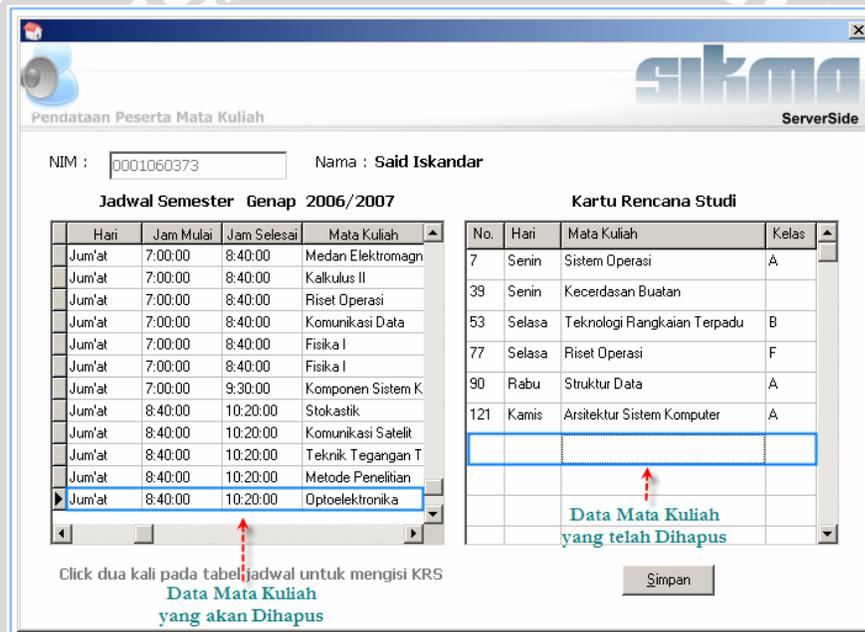
Sumber: *Implementasi*

Administrator mengisi Nomor Induk Mahasiswa (NIM) dengan nomor NIM '0001060373'. Administrator menempatkan posisi kursor pada *record* dengan Mata Kuliah 'Bahasa Assembly' dan menekan tombol kiri pada *mouse* sebanyak dua kali untuk menambah data peserta mata kuliah baru. Administrator menekan tombol *simpan* untuk menyimpan data pada tabel `d_peserta_mk`.

### 5.2.7.2 Implementasi Penghapusan Data Peserta Mata Kuliah

Proses menghapus data peserta mata kuliah dilakukan dengan menghapus data peserta mata kuliah terdapat dalam tabel data peserta mata kuliah. Data peserta mata kuliah yang dihapus akan dihilangkan dari tabel `d_peserta_mk`. Data peserta mata kuliah hanya dapat dihapus oleh administrator.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada tabel Kartu Rencana Studi. Administrator dapat menghapus data peserta mata kuliah dengan menekan tombol kiri *mouse* sebanyak dua kali pada tabel Kartu Rencana Studi (KRS). Implementasi penghapusan data peserta mata kuliah ditunjukkan dalam Gambar 5.28.



**Gambar 5.28** Implementasi Penghapusan Data Peserta Mata Kuliah

Sumber: *Implementasi*

Administrator akan menghapus *record* data Mata Kuliah 'Optoelektronika' dengan menekan tombol kiri *mouse* sebanyak dua kali untuk menghapus *record* data Mata Kuliah 'Optoelektronika' pada tabel Kartu Rencana Studi (KRS).

## 5.2.8 Implementasi Perangkat Lunak untuk Administrasi Data Jadwal Mata Kuliah

Administrasi data jadwal mata kuliah merupakan proses manajemen data jadwal mata kuliah pada tabel `d_jadwal` di basis data kehadiran. Manajemen data jadwal mata kuliah meliputi menambah data, mengubah data, menghapus data, dan mencari data jadwal mata kuliah pada tabel `d_jadwal`.

### 5.2.8.1 Implementasi Penambahan Data Jadwal Mata Kuliah

Proses menambah data jadwal mata kuliah dilakukan dengan memasukkan data jadwal mata kuliah baru ke dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah baru yang dimasukkan akan disimpan dalam tabel `d_jadwal`.

Administrator dapat menambah data jadwal mata kuliah dengan menekan tombol navigasi bertanda (+) dan memasukkan data jadwal mata kuliah baru ke dalam tabel data jadwal mata kuliah. Administrator dapat menyimpan data jadwal mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓). Implementasi penambahan data jadwal mata kuliah ditunjukkan dalam Gambar 5.29.



Gambar 5.29 Implementasi Penambahan Data Jadwal Mata Kuliah  
Sumber: *Implementasi*

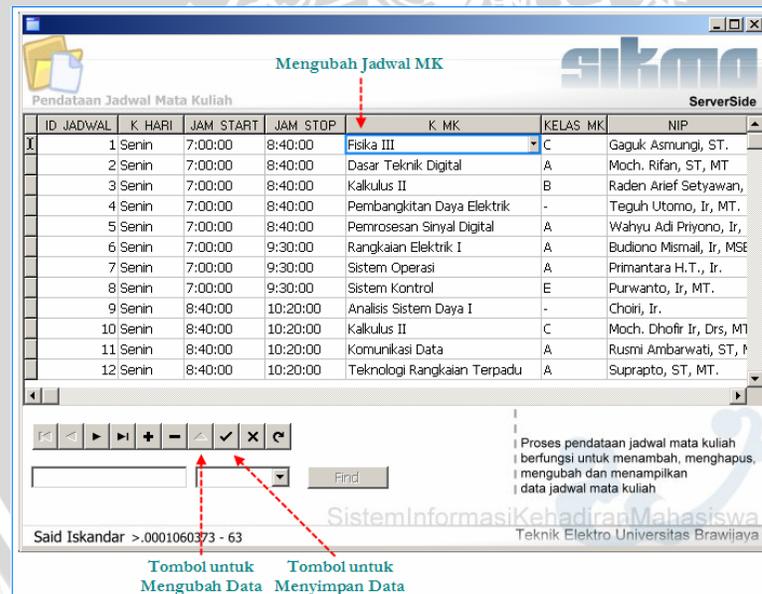
Administrator menempatkan posisi kursor pada *record* terakhir dengan ID\_JADWAL '1' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data jadwal mata kuliah baru. Administrator mengisi *record* kosong dengan ID\_JADWAL '18', K\_HARI 'Senin', JAM\_START '10:20:00',

JAM\_STOP '12:50:00', K\_MK 'Medan Elektromagnetik', KELAS\_MK 'A', dan NIP 'Erfan Achmad Dahlan, Ir.'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel d\_jadwal.

### 5.2.8.2 Implementasi Perubahan Data Jadwal Mata Kuliah

Proses mengubah data jadwal mata kuliah dilakukan dengan mengubah data jadwal mata kuliah yang terdapat dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah yang diubah akan disimpan dalam tabel d\_jadwal.

Administrator dapat mengubah data jadwal mata kuliah dengan menekan tombol navigasi bertanda (▲) dan memasukkan data jadwal mata kuliah yang akan diubah ke dalam *form* pendataan jadwal mata kuliah. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses perubahan data sedang dilakukan. Administrator dapat menyimpan data jadwal mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi perubahan data jadwal mata kuliah ditunjukkan dalam Gambar 5.30.



**Gambar 5.30** Implementasi Perubahan Data Jadwal Mata Kuliah

Sumber: *Implementasi*

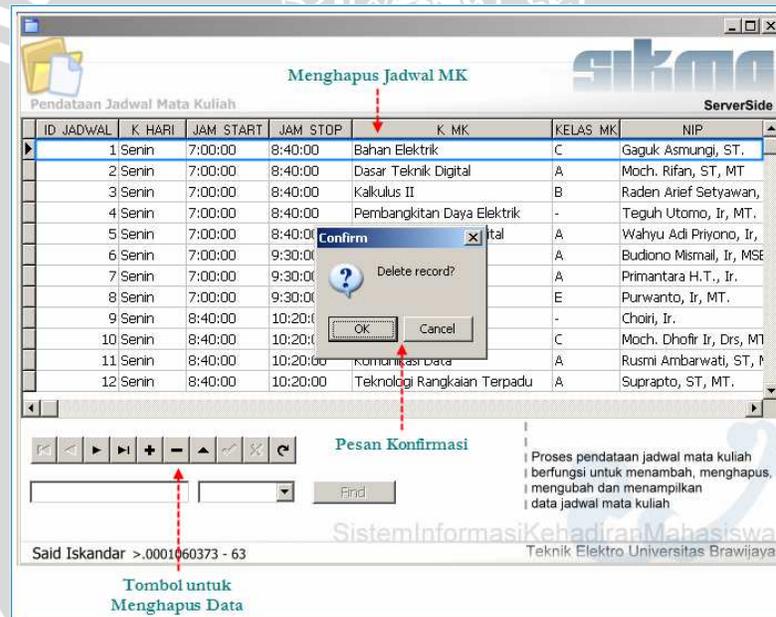
Administrator menempatkan posisi kursor pada *record* pertama dengan K\_MK 'TKE 203' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data jadwal mata kuliah. Administrator mengganti K\_MK 'Bahan Elektrik' dengan K\_MK

‘Fisika III’. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel d\_jadwal.

### 5.2.8.3 Implementasi Penghapusan Data Jadwal Mata Kuliah

Proses menghapus data jadwal mata kuliah dilakukan dengan menghapus data jadwal mata kuliah terdapat dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah yang dihapus akan dihilangkan dari tabel d\_jadwal.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data jadwal mata kuliah dengan menekan tombol navigasi bertanda (−). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus. Implementasi penghapusan data jadwal mata kuliah ditunjukkan dalam Gambar 5.31.

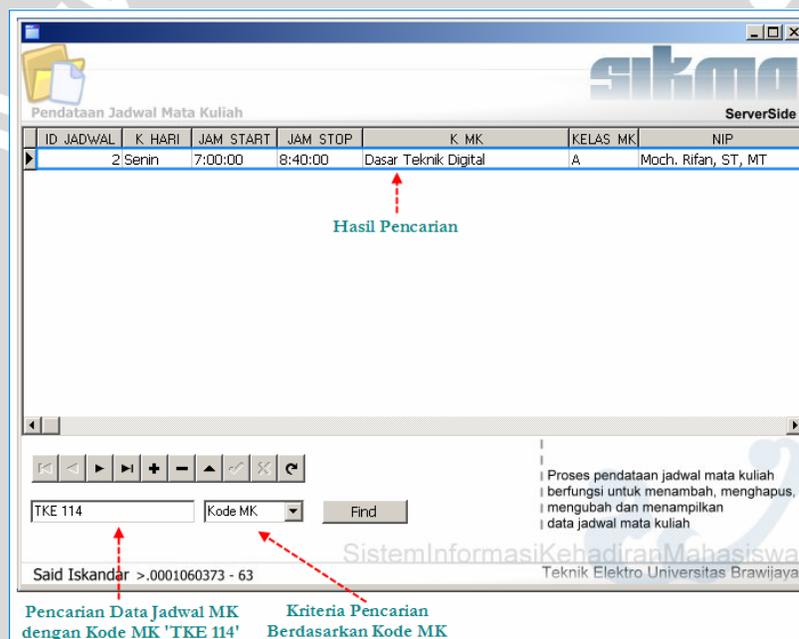


**Gambar 5.31** Implementasi Penghapusan Data Jadwal Mata Kuliah  
Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan ID\_JADWAL ‘1’, K\_HARI ‘Senin’, K\_HARI ‘Senin’, K\_MK ‘Bahan Elektrik’, dan KELAS\_MK ‘C’ dengan menekan tombol navigasi bertanda (−) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel d\_jadwal.

#### 5.2.8.4 Implementasi Pencarian Data Jadwal Mata Kuliah

Proses pencarian data jadwal mata kuliah merupakan proses untuk mencari data jadwal mata kuliah sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian. Administrator menekan tombol cari atau tombol enter untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data jadwal mata kuliah bila ditemukan kesesuaian antara kata kunci pencarian dengan data jadwal mata kuliah pada tabel `d_jadwal`. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian. Implementasi pencarian data jadwal mata kuliah ditunjukkan dalam Gambar 5.32.

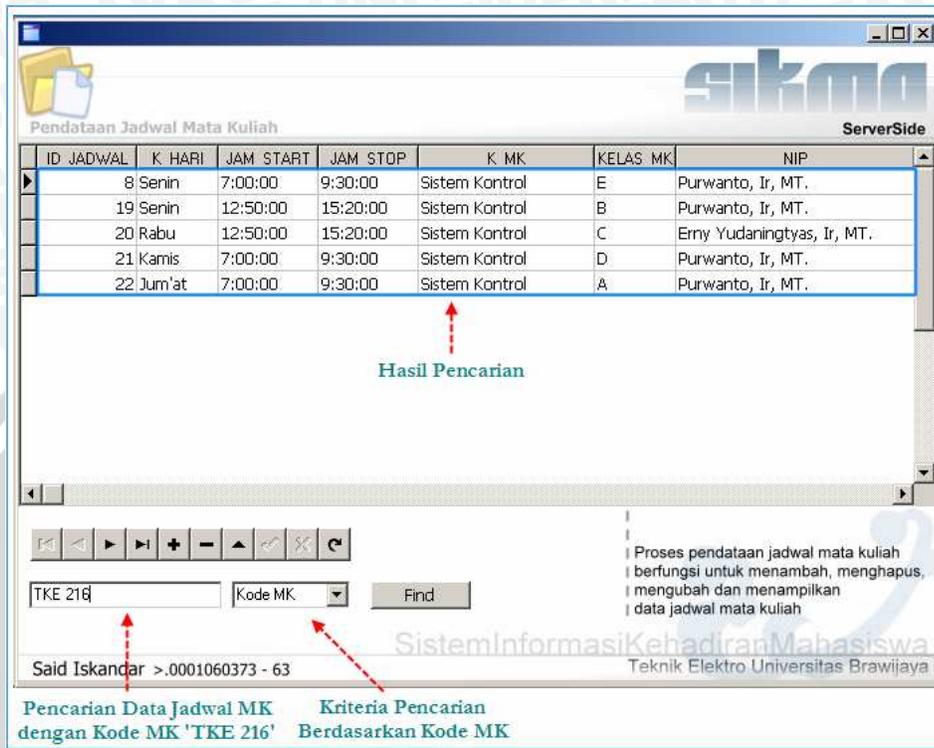


Gambar 5.32 Implementasi Pencarian Data Jadwal Mata Kuliah

Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa `K_MK` dan mengisi kata kunci pencarian dengan `K_MK` 'TKE 114' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan `K_MK` 'TKE 114' menghasilkan keluaran pencarian pada tabel data jadwal mata kuliah berupa *record* data jadwal mata kuliah dengan `ID_JADWAL` '2', `K_HARI` 'Senin', `JAM_START` '7:00:00', `JAM_STOP` '8:40:00', `K_MK` 'Dasar Teknik Digital', `KELAS_MK` 'A', dan `NIP` 'Moch. Rif'an, ST, MT.'.

Implementasi pencarian data jadwal mata kuliah untuk mata kuliah yang mempunyai lebih dari satu kelas ditunjukkan dalam Gambar 5.33.



**Gambar 5.33** Implementasi Pencarian Data Jadwal Mata Kuliah lebih dari satu Kelas  
Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa K\_MK dan mengisi kata kunci pencarian dengan K\_MK 'TKE 216' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan K\_MK 'TKE 216' menghasilkan keluaran pencarian pada tabel data jadwal mata kuliah berupa *record* data jadwal mata kuliah yang ditunjukkan pada Tabel 5.2.

**Tabel 5.2** Hasil Pencarian Data Jadwal Mata Kuliah lebih dari satu Kelas

ID_HADIR	ID_JADWAL	JAM_START	JAM_STOP	K_MK	KELAS	NIP
8	Senin	7:00:00	9:30:00	Sistem Kontrol	E	Purwanto, Ir, MT.
19	Senin	12:50:00	15:20:00	Sistem Kontrol	B	Erny Yudaningtyas, Ir, MT.
20	Rabu	12:50:00	15:20:00	Sistem Kontrol	C	Purwanto, Ir, MT.
21	Kamis	7:00:00	9:30:00	Sistem Kontrol	D	Purwanto, Ir, MT.
22	Jum'at	7:00:00	9:30:00	Sistem Kontrol	A	Purwanto, Ir, MT.

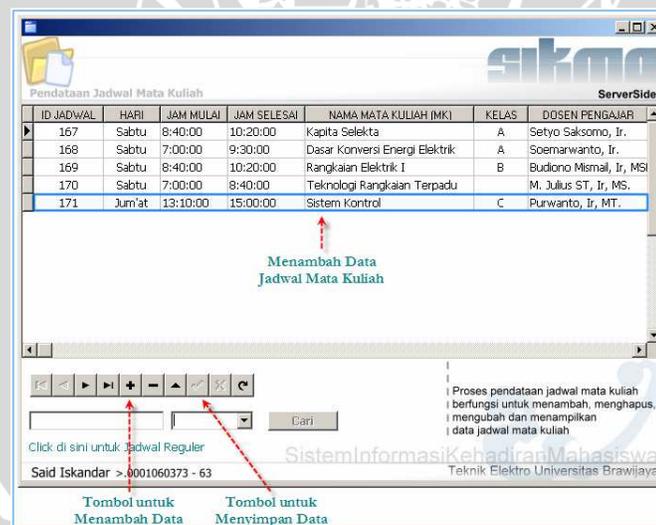
## 5.2.9 Implementasi Perangkat Lunak untuk Administrasi Data Jadwal Mata Kuliah Tambahan

Administrasi data jadwal mata kuliah tambahan merupakan proses manajemen data jadwal mata kuliah pada tabel `d_jadwal` di basis data kehadiran. Manajemen data jadwal mata kuliah tambahan meliputi menambah data, mengubah data, menghapus data, dan mencari data jadwal mata kuliah pada tabel `d_jadwal`.

### 5.2.9.1 Implementasi Penambahan Data Jadwal Mata Kuliah Tambahan

Proses menambah data jadwal mata kuliah tambahan dilakukan dengan memasukkan data jadwal mata kuliah baru ke dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah baru yang dimasukkan akan disimpan dalam tabel `d_jadwal`.

Administrator dapat menambah data jadwal mata kuliah dengan menekan tombol navigasi bertanda (+) dan memasukkan data jadwal mata kuliah baru ke dalam tabel data jadwal mata kuliah. Administrator dapat menyimpan data jadwal mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓). Implementasi penambahan data jadwal mata kuliah tambahan ditunjukkan dalam Gambar 5.34.



Gambar 5.34 Implementasi Penambahan Data Jadwal Mata Kuliah Tambahan  
Sumber: *Implementasi*

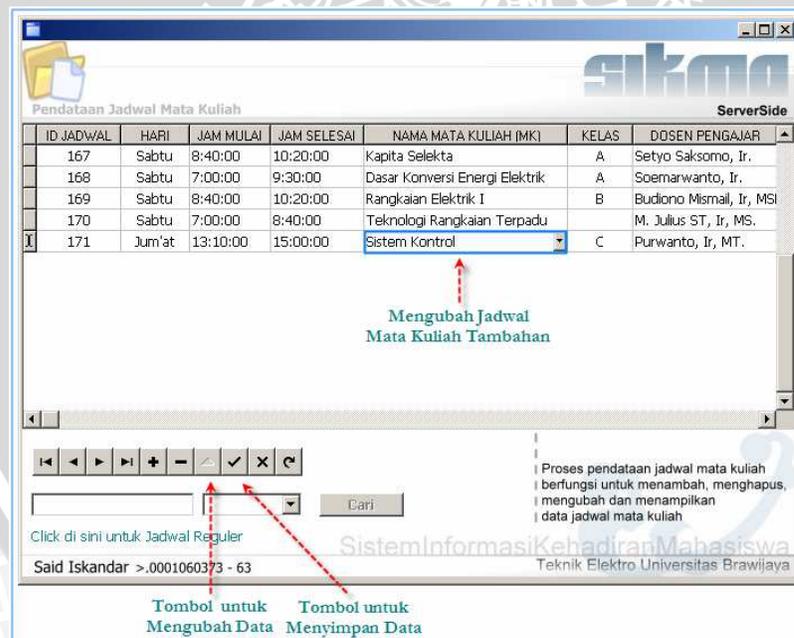
Administrator menempatkan posisi kursor pada *record* terakhir dengan ID\_JADWAL '1' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data jadwal mata kuliah baru. Administrator mengisi *record* kosong dengan ID\_JADWAL '18', K\_HARI 'Senin', JAM\_START '10:20:00',

JAM\_STOP '12:50:00', K\_MK 'Medan Elektromagnetik', KELAS\_MK 'A', dan NIP 'Erfan Achmad Dahlan, Ir.'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel d\_jadwal.

### 5.2.9.2 Implementasi Perubahan Data Jadwal Mata Kuliah Tambahan

Proses mengubah data jadwal mata kuliah tambahan dilakukan dengan mengubah data jadwal mata kuliah yang terdapat dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah yang diubah akan disimpan dalam tabel d\_jadwal.

Administrator dapat mengubah data jadwal mata kuliah tambahan dengan menekan tombol navigasi bertanda (▲) dan memasukkan data jadwal mata kuliah yang akan diubah ke dalam *form* pendataan jadwal mata kuliah. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses perubahan data sedang dilakukan. Administrator dapat menyimpan data jadwal mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi perubahan data jadwal mata kuliah tambahan ditunjukkan dalam Gambar 5.35.



**Gambar 5.35** Implementasi Perubahan Data Jadwal Mata Kuliah Tambahan  
Sumber: *Implementasi*

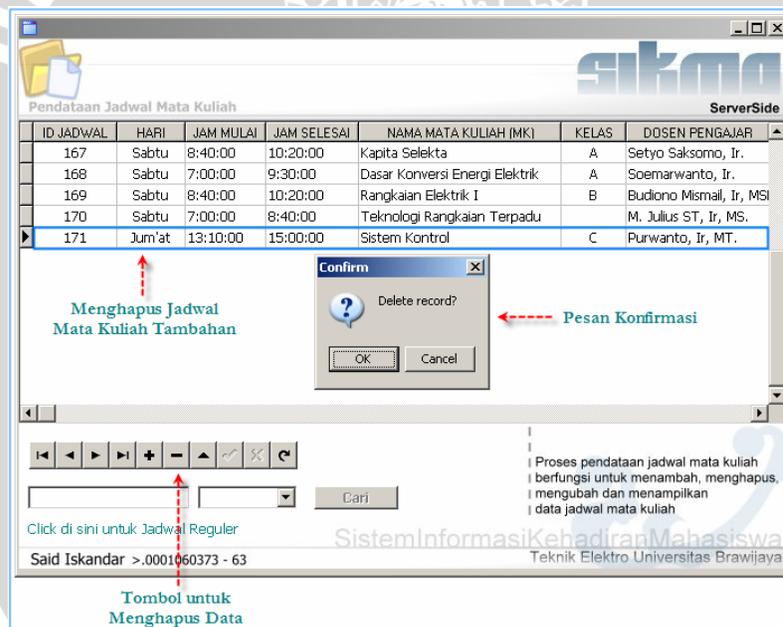
Administrator menempatkan posisi kursor pada *record* pertama dengan K\_MK 'TKE 203' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data jadwal mata kuliah. Administrator mengganti K\_MK 'Bahan Elektrik' dengan K\_MK

‘Fisika III’. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel `d_jadwal`.

### 5.2.9.3 Implementasi Penghapusan Data Jadwal Mata Kuliah Tambahan

Proses menghapus data jadwal mata kuliah tambahan dilakukan dengan menghapus data jadwal mata kuliah terdapat dalam tabel data jadwal mata kuliah. Data jadwal mata kuliah yang dihapus akan dihilangkan dari tabel `d_jadwal`.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data jadwal mata kuliah dengan menekan tombol navigasi bertanda (–). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus. Implementasi penghapusan data jadwal mata kuliah tambahan ditunjukkan dalam Gambar 5.36.

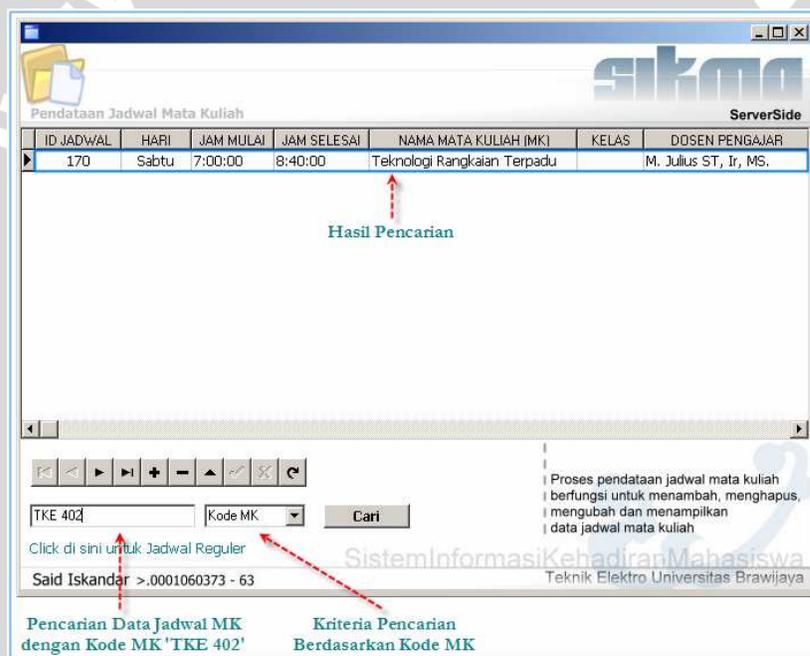


**Gambar 5.36** Implementasi Penghapusan Data Jadwal Mata Kuliah Tambahan  
Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan `ID_JADWAL '1'`, `K_HARI 'Senin'`, `K_HARI 'Senin'`, `K_MK 'Bahan Elektrik'`, dan `KELAS_MK 'C'` dengan menekan tombol navigasi bertanda (–) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `d_jadwal`.

#### 5.2.9.4 Implementasi Pencarian Data Jadwal Mata Kuliah Tambahan

Proses pencarian data jadwal mata kuliah tambahan merupakan proses untuk mencari data jadwal mata kuliah sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian. Administrator menekan tombol cari atau tombol `enter` untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data jadwal mata kuliah bila ditemukan kesesuaian antara kata kunci pencarian dengan data jadwal mata kuliah pada tabel `d_jadwal`. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian. Implementasi pencarian data jadwal mata kuliah tambahan ditunjukkan dalam Gambar 5.37.



**Gambar 5.37** Implementasi Pencarian Data Jadwal Mata Kuliah Tambahan

Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa `K_MK` dan mengisi kata kunci pencarian dengan `K_MK` 'TKE 114' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol `enter` untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan `K_MK` 'TKE 114' menghasilkan keluaran pencarian pada tabel data jadwal mata kuliah berupa *record* data jadwal mata kuliah dengan `ID_JADWAL` '2', `K_HARI` 'Senin', `JAM_START` '7:00:00', `JAM_STOP` '8:40:00', `K_MK` 'Dasar Teknik Digital', `KELAS_MK` 'A', dan `NIP` 'Moch. Rif'an, ST, MT.'.

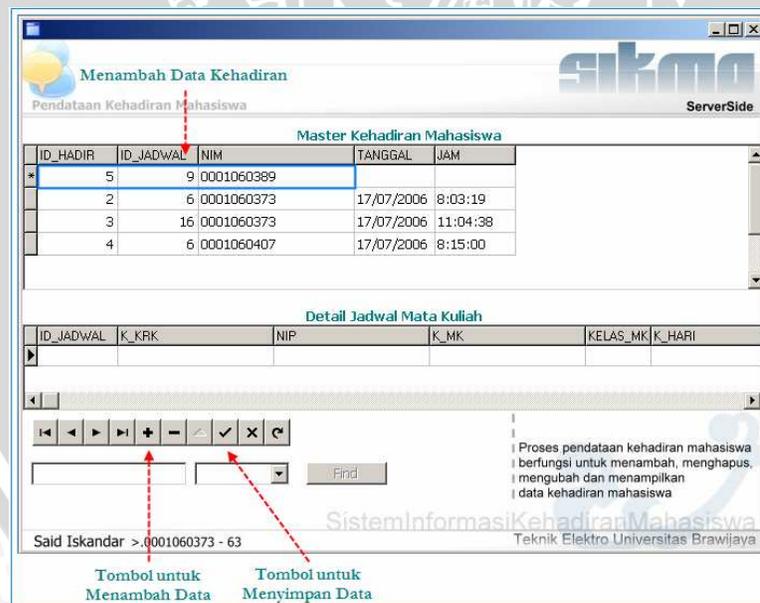
## 5.2.10 Implementasi Perangkat Lunak untuk Administrasi Data Kehadiran Mahasiswa

Administrasi data kehadiran mahasiswa merupakan proses manajemen data kehadiran mahasiswa pada tabel `d_hadir` di basis data kehadiran. Manajemen data kehadiran mahasiswa meliputi menambah data, mengubah data, menghapus data, dan mencari data kehadiran mahasiswa pada tabel `d_hadir`.

### 5.2.10.1 Implementasi Penambahan Data Kehadiran Mahasiswa

Proses menambah data kehadiran mahasiswa dilakukan dengan memasukkan data kehadiran mahasiswa baru ke dalam tabel data kehadiran mahasiswa. Data kehadiran mahasiswa baru yang dimasukkan akan disimpan dalam tabel `d_hadir`.

Administrator dapat menambah data kehadiran mahasiswa dengan menekan tombol navigasi bertanda (+) dan memasukkan data kehadiran mahasiswa baru ke dalam tabel data kehadiran mahasiswa. Administrator dapat menyimpan data kehadiran mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓). Implementasi penambahan data kehadiran mahasiswa ditunjukkan dalam Gambar 5.38.



Gambar 5.38 Implementasi Penambahan Data Kehadiran Mahasiswa  
Sumber: *Implementasi*

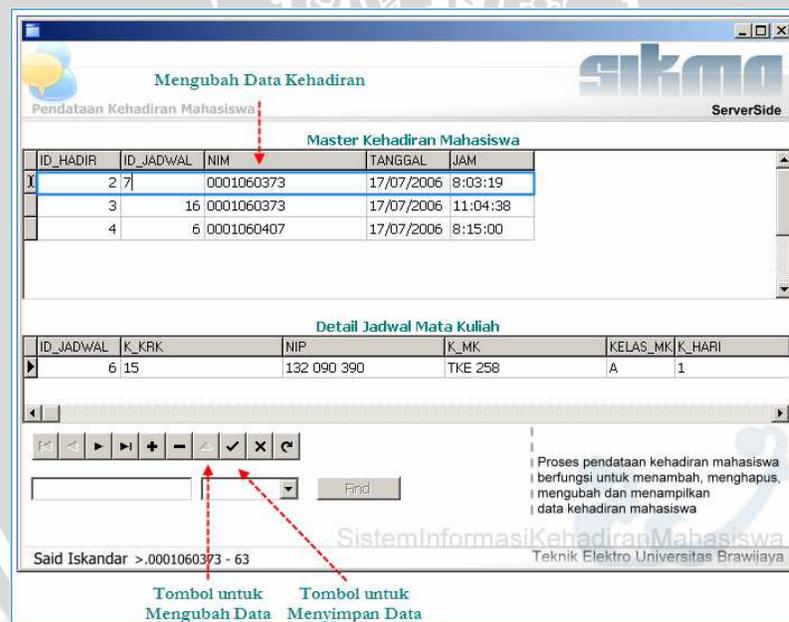
Administrator menempatkan posisi kursor pada *record* pertama dengan ID\_HADIR '2' dan menekan tombol navigasi bertanda (+) untuk menampilkan *record* kosong yang akan ditambahkan data kehadiran baru. Administrator mengisi *record*

kosong dengan ID\_HADIR '5', ID\_JADWAL '9', dan NIM '0001060389'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel d\_hadir.

### 5.2.10.2 Implementasi Pengubahan Data Kehadiran Mahasiswa

Proses mengubah data kehadiran mahasiswa dilakukan dengan mengubah data kehadiran mahasiswa yang terdapat dalam tabel data kehadiran mahasiswa. Data kehadiran mahasiswa yang diubah akan disimpan dalam tabel d\_hadir.

Administrator dapat mengubah data kehadiran mahasiswa dengan menekan tombol navigasi bertanda (▲) dan memasukkan data kehadiran mahasiswa yang akan diubah ke dalam tabel data kehadiran mahasiswa. Tombol navigasi bertanda (▲) menjadi tidak aktif selama proses pengubahan data sedang dilakukan. Administrator dapat menyimpan data kehadiran mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√). Implementasi pengubahan data kehadiran mahasiswa ditunjukkan dalam Gambar 5.39.



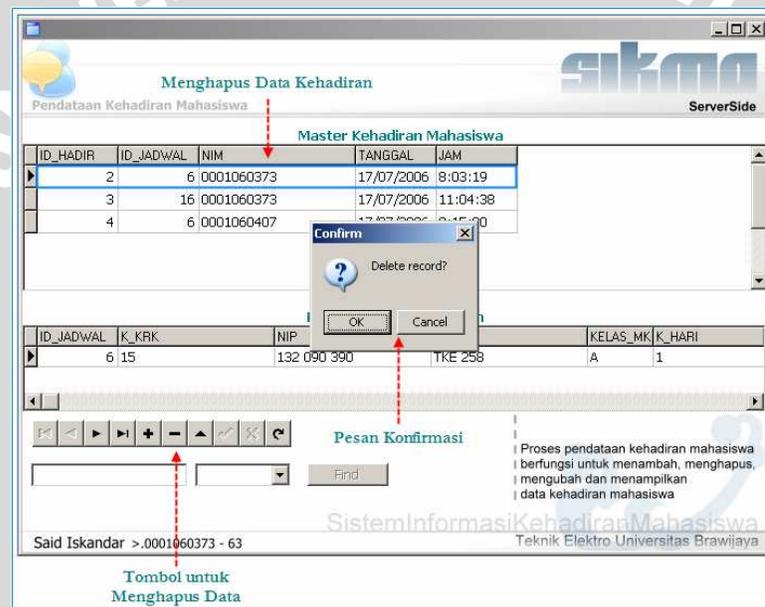
Gambar 5.39 Implementasi Pengubahan Data Kehadiran Mahasiswa  
Sumber: *Implementasi*

Administrator menempatkan posisi kursor pada *record* pertama dengan ID\_HADIR '2' dan menekan tombol navigasi bertanda (▲) untuk mengubah *record* data kehadiran mahasiswa. Administrator mengganti ID\_JADWAL '1' dengan ID\_JADWAL '7'. Administrator menekan tombol navigasi bertanda (√) untuk menyimpan data pada tabel d\_hadir.

### 5.2.10.3 Implementasi Penghapusan Data Kehadiran Mahasiswa

Proses menghapus data kehadiran mahasiswa dilakukan dengan menghapus data kehadiran mahasiswa terdapat dalam tabel data kehadiran mahasiswa. Data kehadiran mahasiswa yang dihapus akan dihilangkan dari tabel `d_hadir`.

Administrator menempatkan kursor pada *record* data yang akan dihapus sehingga pada kolom pertama tabel data akan tampil tanda (▶) atau tanda (I). Administrator dapat menghapus data kehadiran mahasiswa dengan menekan tombol navigasi bertanda (–). Administrator akan mendapat pesan konfirmasi untuk memastikan data yang akan dihapus. Implementasi penghapusan data kehadiran mahasiswa ditunjukkan dalam Gambar 5.40.



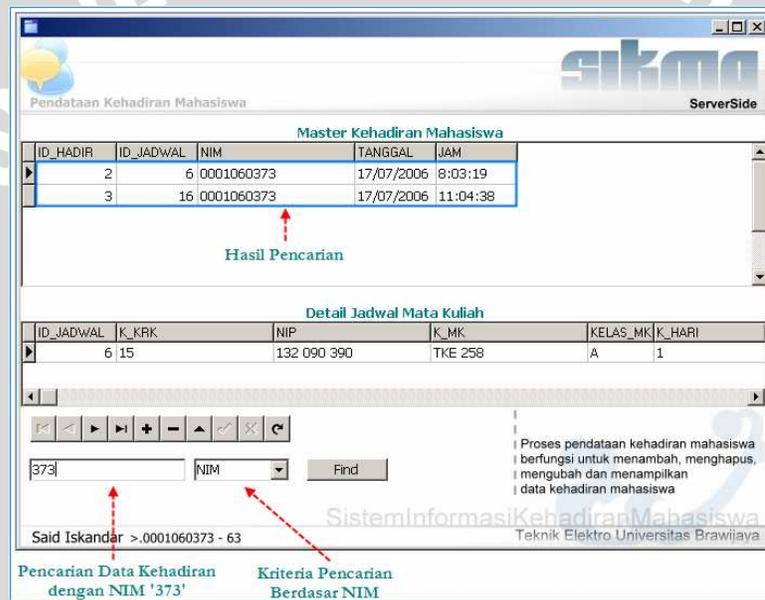
Gambar 5.40 Implementasi Penghapusan Data Kehadiran Mahasiswa

Sumber: *Implementasi*

Administrator akan menghapus *record* data dengan ID\_JADWAL '2', ID\_JADWAL '6', NIM '0001060373', TANGGAL '17/17/2006', dan JAM '8:03:19' dengan menekan tombol navigasi bertanda (–) untuk menghapus *record* data. Pesan konfirmasi akan ditampilkan untuk menanyakan apakah memang benar bahwa *record* data akan dihapus. Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `d_hadir`.

### 5.2.10.4 Implementasi Pencarian Data Kehadiran Mahasiswa

Proses pencarian data kehadiran mahasiswa merupakan proses untuk mencari data kehadiran mahasiswa sesuai dengan kehendak administrator. Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian. Administrator menekan tombol cari atau tombol `enter` untuk menampilkan hasil pencarian. Hasil pencarian akan ditampilkan pada tabel data kehadiran mahasiswa bila ditemukan kesesuaian antara kata kunci pencarian dengan data kehadiran mahasiswa pada tabel `d_hadir`. Hasil pencarian juga menampilkan data yang mengandung kata kunci pencarian. Kata kunci pencarian dapat berupa bagian kata atau seluruh kata sesuai dengan kriteria pencarian. Implementasi pencarian data kehadiran mahasiswa ditunjukkan dalam Gambar 5.41.



Gambar 5.41 Implementasi Pencarian Data Kehadiran Mahasiswa

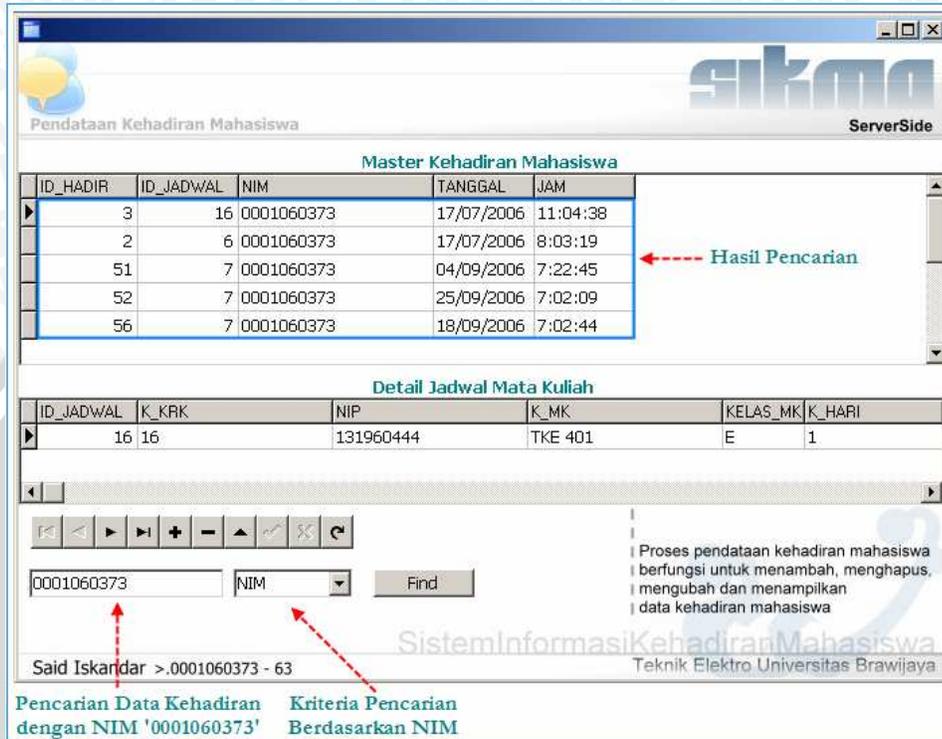
Sumber: *Implementasi*

Administrator memilih kriteria pencarian berupa NIM dan mengisi kata kunci pencarian dengan NIM '373' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol `enter` untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan NIM '373' menghasilkan keluaran pencarian pada tabel data kehadiran mahasiswa berupa *record* data kehadiran mahasiswa yang ditunjukkan pada Tabel 5.3.

Tabel 5.3 Hasil Pencarian Data Kehadiran Mahasiswa

ID_HADIR	ID_JADWAL	NIM	TANGGAL	JAM
2	6	0001060373	17/07/2006	8:03:19
3	16	0001060373	17/07/2006	11:04:38

Contoh hasil pencarian data kehadiran mahasiswa yang ke dua dengan jadwal mata kuliah dan tanggal yang berbeda. Implementasi pencarian data kehadiran mahasiswa ditunjukkan dalam Gambar 5.42.



**Gambar 5.42** Implementasi Pencarian Data Kehadiran Mahasiswa dengan Jadwal dan Tanggal berbeda  
 Sumber: *Implementasi*

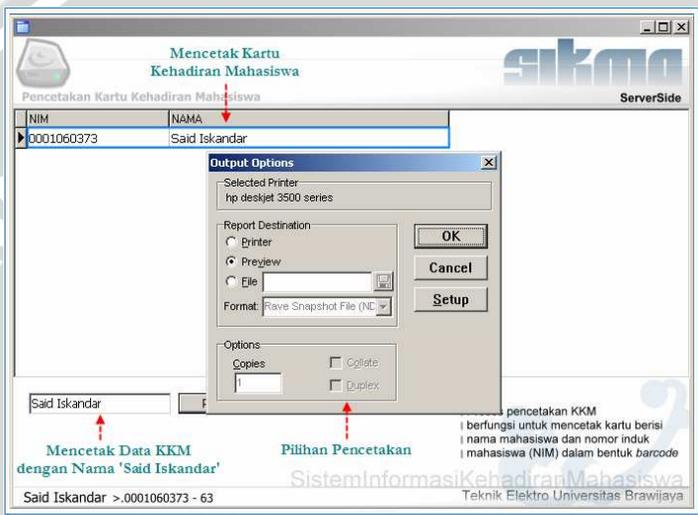
Administrator memilih kriteria pencarian berupa NIM dan mengisi kata kunci pencarian dengan NIM '0001060373' pada masukan pencarian. Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian. Kata kunci pencarian dengan NIM '0001060373' menghasilkan keluaran pencarian pada tabel data kehadiran mahasiswa berupa *record* data kehadiran mahasiswa yang ditunjukkan pada Tabel 5.4.

**Tabel 5.4** Hasil Pencarian Data Kehadiran Mahasiswa dengan Tanggal yang berbeda

ID_HADIR	ID_JADWAL	NIM	TANGGAL	JAM
3	16	0001060373	17/07/2006	11:04:38
2	6	0001060373	17/07/2006	8:03:19
51	7	0001060373	04/09/2006	7:22:45
52	7	0001060373	25/09/2006	7:02:09
56	7	0001060373	18/09/2006	7:02:44

### 5.2.11 Implementasi Perangkat Lunak untuk Pencetakan Kartu Kehadiran Mahasiswa (KKM)

Proses pencetakan KKM berfungsi untuk mencetak kartu berisi nama mahasiswa, Nomor Induk Mahasiswa (NIM), dan NIM dalam bentuk *barcode* berdasarkan tabel *m\_mhs*. NIM mahasiswa yang diperoleh dari tabel *m\_mhs* akan ditampilkan pada *form* pencetakan KKM. Administrator akan diberikan pilihan, yaitu mencetak KKM (*print*), melihat tampilan KKM (*preview*), atau menyimpan KKM dalam *file*. Implementasi pencetakan KKM ditunjukkan dalam Gambar 5.43.



Gambar 5.43 Implementasi Pencetakan Kartu Kehadiran Mahasiswa  
Sumber: Implementasi

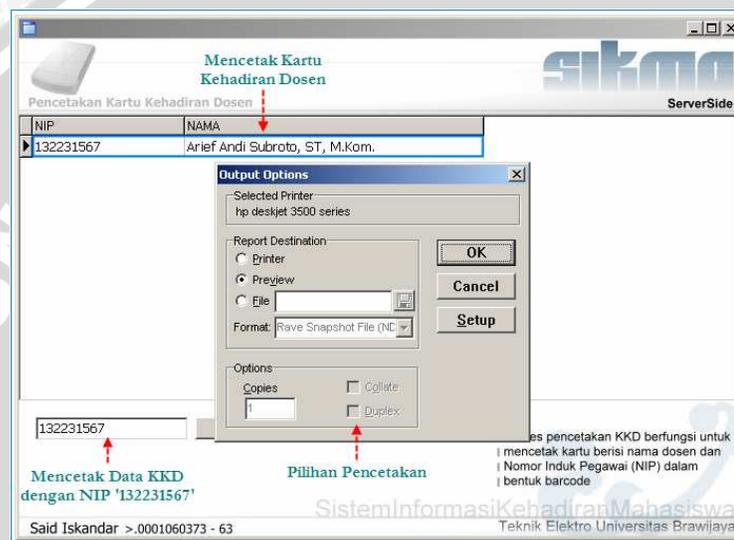
Administrator mengisi kata kunci pencarian dengan NAMA 'Said Iskandar' dan *form* pilihan pencetakan akan ditampilkan setelah administrator menekan tombol Print. Administrator dapat memilih untuk mencetak ke *printer*, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk *file*. Implementasi kartu kehadiran mahasiswa ditunjukkan dalam Gambar 5.44.



Gambar 5.44 Implementasi Kartu Kehadiran Mahasiswa  
Sumber: Implementasi

### 5.2.12 Implementasi Perangkat Lunak untuk Pencetakan Kartu Kehadiran Dosen (KKD)

Proses pencetakan KKD berfungsi untuk mencetak kartu berisi nama dosen, Nomor Induk Dosen (NIP), dan NIP dalam bentuk *barcode* berdasarkan tabel `m_dosen`. NIP dosen yang diperoleh dari tabel `m_dosen` akan ditampilkan pada *form* pencetakan KKD. Administrator akan diberikan pilihan, yaitu mencetak KKD (*print*), melihat tampilan KKD (*preview*), atau menyimpan KKD dalam *file*. Implementasi pencetakan KKD ditunjukkan dalam Gambar 5.45.



**Gambar 5.45** Implementasi Pencetakan Kartu Kehadiran Dosen

Sumber: *Implementasi*

Administrator mengisi kata kunci pencarian dengan NIP '132231567' dan *form* pilihan pencetakan akan ditampilkan setelah administrator menekan tombol `Print`. Administrator dapat memilih untuk mencetak ke *printer*, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk *file*. Implementasi kartu kehadiran dosen ditunjukkan dalam Gambar 5.46.

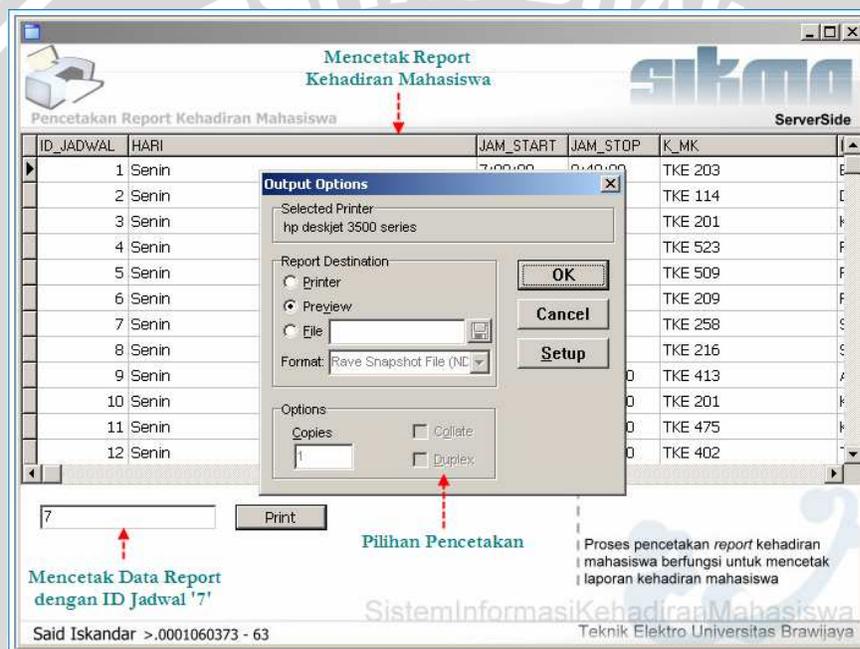


**Gambar 5.46** Implementasi Kartu Kehadiran Dosen

Sumber: *Implementasi*

### 5.2.13 Implementasi Perangkat Lunak untuk Pencetakan *Report* Kehadiran Mahasiswa

Proses cetak *report* kehadiran mahasiswa merupakan proses mencetak *report* kehadiran mahasiswa berdasarkan tabel `d_hadir`. Data jadwal mata kuliah akan ditampilkan pada *form* pencetakan *report* kehadiran mahasiswa. Administrator dapat mencetak *report* kehadiran mahasiswa berdasarkan id jadwal mata kuliah. Administrator akan diberikan pilihan, yaitu mencetak *report* (*print*), melihat tampilan *report* (*preview*), atau menyimpan *report* dalam *file*. Implementasi pencetakan *report* kehadiran mahasiswa ditunjukkan dalam Gambar 5.47.



Gambar 5.47 Implementasi Pencetakan *Report* Kehadiran Mahasiswa

Sumber: *Implementasi*

Administrator mengisi kata kunci pencarian dengan ID\_JADWAL '7' dan *form* pilihan pencetakan akan ditampilkan setelah administrator menekan tombol `Print`. Administrator dapat memilih untuk mencetak ke *printer*, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk *file*.

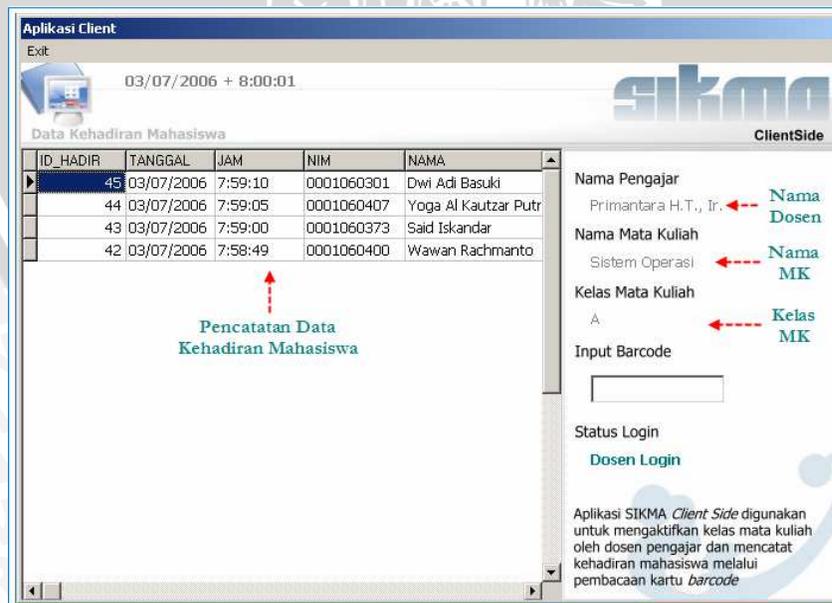
Implementasi *report* kehadiran mahasiswa ditunjukkan dalam Gambar 5.48.

NIM	Nama Mahasiswa	Jumlah Hadir	%
0001060272	Ahmad Wildan Nuruzzaman	2	50
0001060301	Dwi Adi Basuki	1	25
0001060373	Said Iskandar	3	75
0001060400	Wawan Rachmanto	4	100
0001060407	Yoga Al Kautzar Putra	3	75

**Gambar 5.48** Implementasi *Report* Kehadiran Mahasiswa  
Sumber: *Implementasi*

### 5.3 Implementasi Antarmuka Aplikasi *Client*

Aplikasi *client* berfungsi untuk proses *login* dosen dan pencatatan kehadiran mahasiswa. Aplikasi *client* dipilah berdasarkan pengguna, yaitu dosen dan mahasiswa. Dosen melakukan otentifikasi *login* untuk pengecekan jadwal mengajar mata kuliah dan kelas mata kuliah. Mahasiswa melakukan pencatatan kehadiran setelah melalui pengecekan daftar peserta mata kuliah. Implementasi antarmuka aplikasi *client* ditunjukkan dalam Gambar 5.49.



**Gambar 5.49** Implementasi Antarmuka Kehadiran Mahasiswa  
Sumber: *Implementasi*

Dosen memulai proses *login* dengan mengisi *input barcode* dengan kartu kehadiran dosen yang akan dibaca oleh *barcode scanner*. Implementasi login yang dilakukan oleh dosen pada aplikasi client ditunjukkan dalam Gambar 5.50.



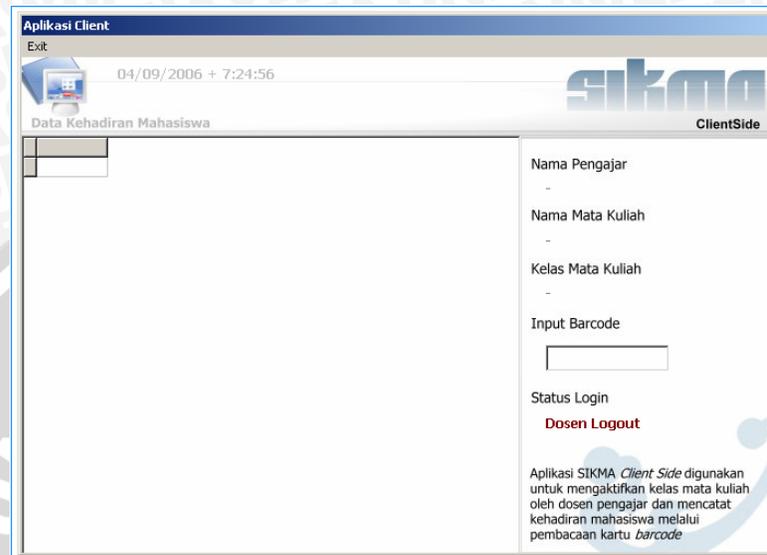
**Gambar 5.50** Implementasi Proses *Login* Dosen  
 Sumber: *Implementasi*

Aplikasi *client* akan menampilkan Nama Pengajar, Nama Mata Kuliah, dan Kelas Mata Kuliah bila dosen tersebut *login* sesuai dengan hari dan jam sesuai dengan jadwal mengajar dosen yang bersangkutan. Implementasi hasil proses *login* dosen ditunjukkan dalam Gambar 5.51.



**Gambar 5.51** Implementasi Hasil Proses *Login* Dosen  
 Sumber: *Implementasi*

Dosen dapat melakukan proses *logout* dengan mengisi *input barcode* dengan kartu kehadiran dosen yang dibaca oleh *barcode scanner* untuk mengakhiri proses pencatatan kehadiran mahasiswa. Proses *logout* yang berhasil dilakukan oleh dosen ditunjukkan dalam Gambar 5.52.



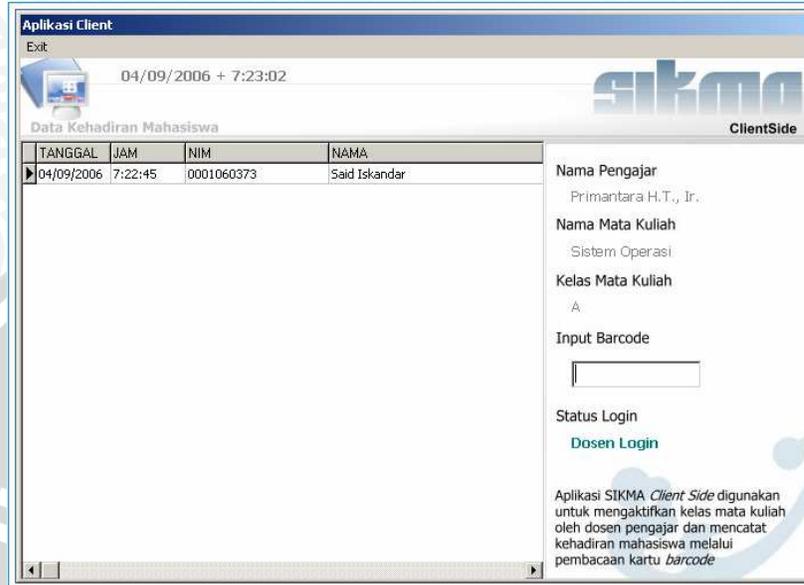
**Gambar 5.52** Proses *logout* dosen berhasil dilakukan  
Sumber: Implementasi

Mahasiswa dapat memulai proses pencatatan kehadiran setelah dosen melakukan proses *login*. Mahasiswa mengisi *input barcode* dengan kartu kehadiran mahasiswa yang akan dibaca oleh *barcode scanner*. Halaman antarmuka aplikasi *client* untuk proses pencatatan kehadiran ditunjukkan dalam Gambar 5.53.



**Gambar 5.53** Halaman Antarmuka Aplikasi *Client* untuk Proses Pencatatan Kehadiran  
Sumber: Implementasi

Mahasiswa dapat dicatat kehadirannya bila mahasiswa tersebut telah terdaftar pada mata kuliah yang sesuai dengan proses *login* dosen. Tampilan antarmuka hasil pencatatan kehadiran mahasiswa yang ditunjukkan dalam Gambar 5.54.



**Gambar 5.54** Halaman Antarmuka Hasil Pencatatan Kehadiran Mahasiswa  
Sumber: *Implementasi*

Aplikasi *client* akan menampilkan pesan peringatan bila pengguna melakukan kesalahan dalam proses *login* dosen dan pencatatan kehadiran mahasiswa. Pesan peringatan aplikasi *client* terdiri dari empat bagian, yaitu:

1. Pesan peringatan akan ditampilkan bila dosen melakukan kesalahan dalam memasukkan *login* atau tidak terdaftar dalam tabel *m\_dosen* pada basis data kehadiran. Pesan kesalahan dalam proses *login* dosen ditunjukkan dalam Gambar 5.55.



**Gambar 5.55** Implementasi Pesan Peringatan *Login*  
Sumber: *Implementasi*

2. Pesan peringatan akan ditampilkan bila proses *login* dosen berhasil tetapi dosen yang bersangkutan tidak mempunyai jadwal mengajar pada hari dan jam saat aplikasi *client* dijalankan. Pesan kesalahan dalam pengaktifan jadwal mengajar ditunjukkan dalam Gambar 5.56.



**Gambar 5.56** Implementasi Pesan Peringatan Jadwal Mengajar  
Sumber: *Implementasi*

3. Pesan peringatan akan ditampilkan bila mahasiswa tidak terdaftar pada mata kuliah yang aktif pada saat aplikasi *client* dijalankan sesuai dengan tabel `d_peserta_mk`. Pesan kesalahan dalam pengaktifan mata kuliah ditunjukkan dalam Gambar 5.57.



**Gambar 5.57** Implementasi Pesan Peringatan Mata Kuliah  
Sumber: *Implementasi*

4. Pesan peringatan akan ditampilkan bila mahasiswa melakukan proses pencatatan kehadiran lebih dari satu kali. Pesan kesalahan dalam proses pencatatan kehadiran mahasiswa ditunjukkan dalam Gambar 5.58.

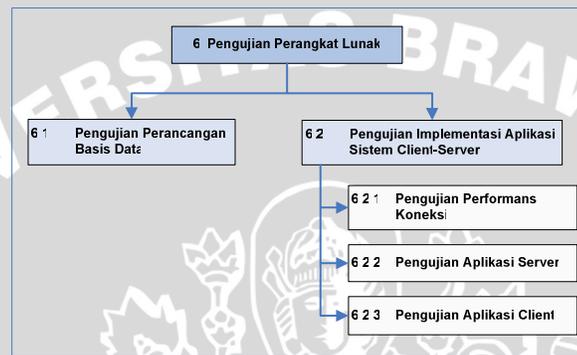


**Gambar 5.58** Implementasi Pesan Peringatan Pencatatan Kehadiran  
Sumber: *Implementasi*

## BAB VI

### PENGUJIAN DAN ANALISIS PERANGKAT LUNAK

Bab ini membahas mengenai pengujian dan analisis terhadap implementasi sistem informasi kehadiran mahasiswa dengan menggunakan *barcode*. Pengujian yang dilakukan meliputi pengujian perancangan basis data dan pengujian implementasi aplikasi sistem secara *client-server*. Diagram pohon pengujian dan analisis perangkat lunak ditunjukkan dalam Gambar 6.1



**Gambar 6.1** Diagram Pohon Pengujian dan Analisis Perangkat Lunak  
Sumber: *Perancangan*

#### 6.1 Pengujian Perancangan Basis Data

Pengujian perancangan basis data bertujuan untuk mengetahui apakah implementasi perancangan basis data yang dilakukan telah sesuai dengan *entity relationship diagram*. Pengujian perancangan basis data dilakukan dengan menggunakan perangkat lunak Sybase PowerDesigner 10.

Pengujian meliputi pembuatan tabel *m\_mhs*, *m\_dosen*, *m\_mk*, *m\_krk*, *m\_login*, *m\_hari*, *d\_peserta\_mk*, *d\_jadwal*, dan *d\_hadir* pada basis data kehadiran.

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses pembuatan basis data kehadiran dan pemeriksaan validitas rancangan tabel yang dilakukan telah sesuai dengan *entity relationship diagram*.

##### B. Spesifikasi dan Konfigurasi Komputer

- Prosesor AMD Duron ~1300 MHz dengan memori 512 MB DDR-SDRAM.
- Sistem operasi Microsoft Windows XP Professional.

**C. Software Aplikasi**

- Sybase PowerDesigner 10
- Server basis data Borland Interbase versi 6.5
- IBEasy+ versi 1.4.6

**D. Prosedur Pengujian**

- Window command prompt *dijalankan dengan memberikan perintah sebagai berikut:*

Start | Run... | Open: cmd.exe

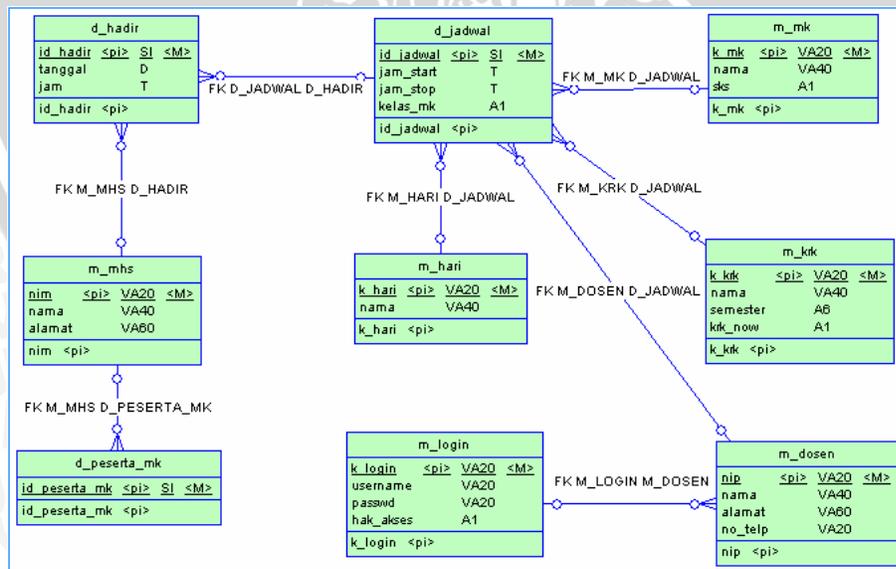
- Server basis data Borland Interbase *dijalankan dengan memberikan perintah sebagai berikut:*

C:\Program Files\Borland\Interbase\bin\ibserver.exe

- Sybase Power Designer 10 *dijalankan dengan memilih menu pada start button windows sebagai berikut:*

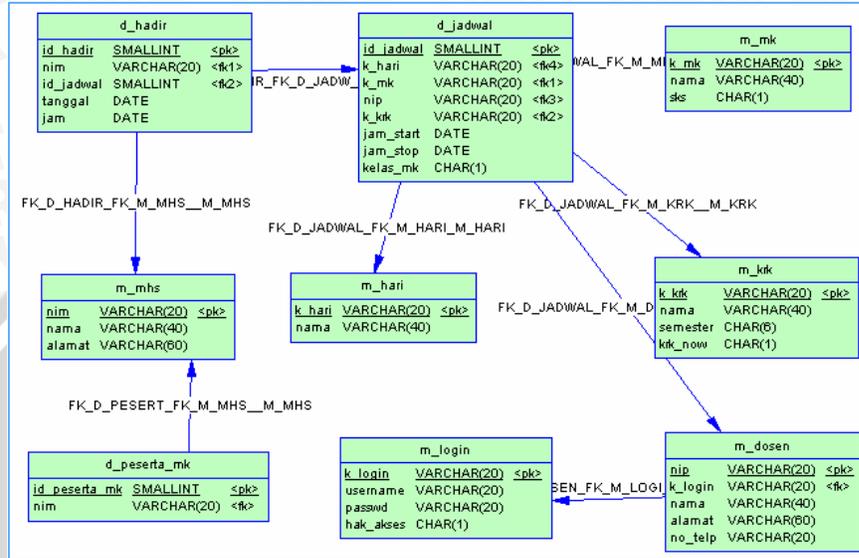
Start | All Programs | Sybase | PowerDesigner 10 | PowerDesigner

- Entity relationship diagram *dari basis data kehadiran digambarkan kembali pada area kerja Conceptual Data Model (CDM).*
- Entity relationship diagram *diperiksa kembali dengan cara menekan tombol Check Model pada toolbar. Hasil pemeriksaan ini disebut dengan CDM Object yang ditunjukkan dalam Gambar 6.2.*



**Gambar 6.2** Conceptual Data Model Object untuk Basis Data Kehadiran  
Sumber: Pengujian

- CDM *object* kemudian diubah menjadi *Physical Data Model (PDM) object* dengan menekan *link Generate Physical Data Model* pada *toolbar tools*. PDM *object* ditunjukkan dalam Gambar 6.3.



Gambar 6.3 Physical Data Model Object untuk Basis Data Kehadiran  
Sumber: Pengujian

### E. Hasil Pengujian dan Analisis

- Script Generate Database dari basis data kehadiran dan hasil proses Generate Database dari PDM Object ditunjukkan dalam Gambar 6.4.

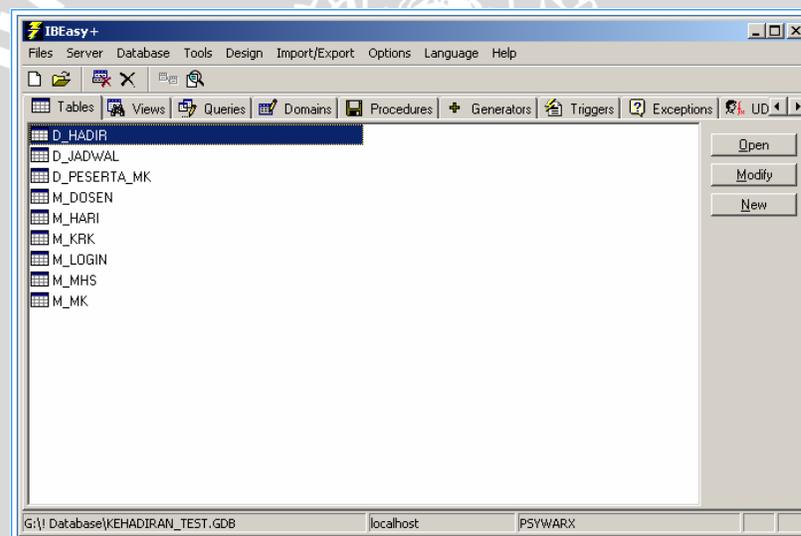
```

Database Generation
Generation: Check model starting...
Generation: Check model successful.
Sorting objects...
Sort completed.
Script Generation...
Creating Tables...
-> Table: d_hadir (D_HADIR)
-> Table: d_jadwal (D_JADWAL)
-> Table: d_peserta_mk (D_PESERTA_MK)
-> Table: m_dosen (M_DOSEN)
-> Table: m_hari (M_HARI)
-> Table: m_krk (M_KRK)
-> Table: m_login (M_LOGIN)
-> Table: m_mhs (M_MHS)
-> Table: m_mk (M_MK)
Creating References...
-> Reference: FK D_JADWAL D_HADIR (FK D_JADWAL D_HADIR)
-> Reference: FK M_MHS D_HADIR (FK M_MHS D_HADIR)
-> Reference: FK M_DOSEN D_JADWAL (FK M_DOSEN D_JADWAL)
-> Reference: FK M_HARI D_JADWAL (FK M_HARI D_JADWAL)
-> Reference: FK M_KRK D_JADWAL (FK M_KRK D_JADWAL)
-> Reference: FK M_MK D_JADWAL (FK M_MK D_JADWAL)
-> Reference: FK M_MHS D_PESERTA_MK (FK M_MHS D_PESERTA_MK)
-> Reference: FK M_LOGIN M_DOSEN (FK M_LOGIN M_DOSEN)
Script Generation completed
Generation successful
    
```

Gambar 6.4 Script Generate Database dari Basis Data Kehadiran  
Sumber: Pengujian

*Script generate database* dari basis data kehadiran dan hasil proses *generate database* dari PDM *object* ditunjukkan dalam lampiran A.

- Sybase PowerDesigner 10 dapat digunakan untuk pembuatan basis data kehadiran dan pemeriksaan validitas tabel sesuai dengan *entity relationship diagram*.
- Basis data kehadiran hasil proses *generate database* dari PDM *object* sesuai dengan hasil implementasi basis data kehadiran pada Borland Interbase.
- Basis data kehadiran hasil proses *generate database* dari PDM *object* terdiri dari tabel *m\_mhs*, *m\_dosen*, *m\_mk*, *m\_krk*, *m\_login*, *m\_hari*, *d\_peserta\_mk*, *d\_jadwal*, dan *d\_hadir*.
- Hasil pembuatan basis data kehadiran menggunakan Sybase PowerDesigner 10 ditunjukkan dalam Gambar 6.5.



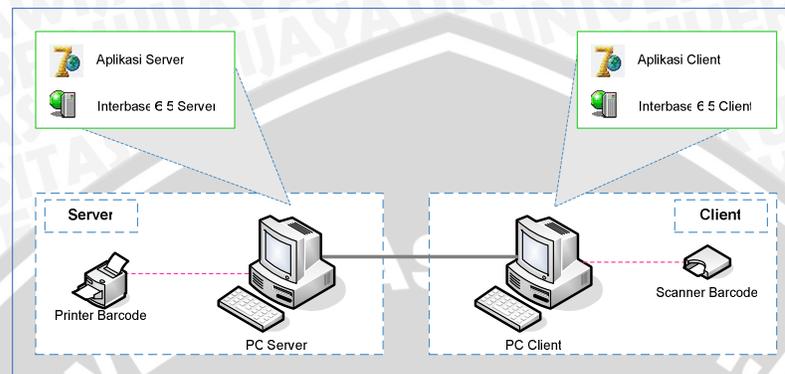
**Gambar 6.5** Basis Data Kehadiran Hasil Proses Generate Database pada Sybase PowerDesigner10  
Sumber: Pengujian

## 6.2 Pengujian Implementasi Aplikasi Sistem Secara *Client-Server*

Pengujian implementasi aplikasi sistem secara *client-server* dilakukan untuk mengetahui apakah Aplikasi Sistem Informasi Kehadiran Mahasiswa dapat dijalankan secara *client-server*.

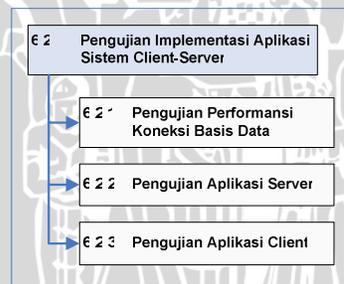
Pengujian implementasi aplikasi sistem secara *client-server* dilakukan dengan menggunakan aplikasi *server* dan Borland Interbase Server (versi 6.5) yang ditempatkan pada komputer *server*. Aplikasi *client* dan Borland Interbase Client yang ditempatkan pada komputer *client*. Administrator mengakses Sistem Informasi

Kehadiran Mahasiswa melalui aplikasi *server* dari komputer *server*. Dosen dan mahasiswa mengakses Sistem Informasi Kehadiran Mahasiswa melalui aplikasi *client* dari komputer *client*. Pengujian implementasi disusun dengan komputer *server* dan *client* dihubungkan dengan jaringan komputer yang ditunjukkan dalam Gambar 6.6.



**Gambar 6.6** Diagram Blok Pengujian secara *Client-Server*  
Sumber: *Perancangan*

Pengujian implementasi aplikasi sistem secara *client-server* yang dilakukan terdiri dari pengujian performansi koneksi basis data, pengujian aplikasi *server*, dan pengujian aplikasi *client*. Diagram pohon pengujian implementasi aplikasi secara *client-server* ditunjukkan dalam Gambar 6.7.



**Gambar 6.7** Diagram Pohon Pengujian Implementasi Aplikasi secara *Client-Server*  
Sumber: *Perancangan*

Pengujian implementasi aplikasi sistem secara *client-server* menggunakan spesifikasi dan konfigurasi komputer sebagai berikut :

A. PC Server:

- Prosesor AMD Duron ~ 1300 MHz dengan memori 512 MB DDR-SDRAM.
- Sistem operasi Microsoft Windows XP Professional.
- Alamat IP pada perangkat Ethernet: 172.17.68.249 *Version* 2002 SP1.
- Printer HP Deskjet 3535.
- *Scanner barcode* Zebex Type Z3080.

**B. PC Client:**

- Prosesor Intel Pentium III – 863 MHz dengan memori 256 MB DDR.
- Sistem operasi Microsoft Windows XP Professional *Version* 2002 SP1.
- Alamat IP pada perangkat ethernet: 172.17.68.242.
- *Scanner barcode* Zebex tipe Z3080.

Perangkat lunak aplikasi yang digunakan pada pengujian implementasi aplikasi sistem secara *client-server* adalah :

**A. PC Server:**

- Borland Interbase Server (versi 6.5)
- IBEasy (versi 1.4.6)
- Active Ports (versi 1.4)

**B. PC Client:**

- Borland Interbase Client
- Aplikasi Client
- MiTeC Interbase Query

**6.2.1 Pengujian Performansi Koneksi**

Pengujian performansi koneksi terdiri dari pengujian koneksi basis data dan pengujian waktu akses *query*. Pengujian koneksi basis data dilakukan untuk mengetahui koneksi yang dilakukan komputer *client* terhadap basis data Borland Interbase pada komputer *server*. Pengujian waktu akses *query* dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan sebuah *query* pada basis data kehadiran.

**6.2.1.1 Pengujian Koneksi Basis Data****A. Tujuan**

Pengujian dilakukan untuk mengetahui koneksi *server* basis data Borland Interbase yang terletak di komputer *server* dalam sebuah jaringan LAN (*Local Area Network*).

**B. Prosedur Pengujian****PC Server:**

- *Window command prompt* dijalankan dengan memberikan perintah sebagai berikut:

Start | Run... | Open: cmd.exe

- *Server* basis data Borland Interbase dijalankan dengan memberikan perintah sebagai berikut:

C:\Program Files\Borland\Interbase\bin\ibserver.exe

- Aplikasi dan status koneksi yang sedang aktif dapat ditampilkan dengan menggunakan aplikasi Active Ports:

C:\Program Files\Active Ports\aports.exe

**PC Client:**

- *Window command prompt* dijalankan dengan memberikan perintah sebagai berikut:

Start | Run... | Open: cmd.exe

- Aplikasi *client* dijalankan dengan memanggil:

G:\! Delphi\delphi\_6\client.exe

- Aplikasi dan status koneksi yang sedang aktif dapat ditampilkan dengan menggunakan aplikasi Active Ports:

C:\Program Files\Active Ports\aports.exe

**C. Hasil Pengujian dan Analisis**

- Proses yang dijalankan pada komputer *server* sebelum ada koneksi dengan komputer *client* ditunjukkan dalam Gambar 6.8. Perintah tersebut digunakan untuk menampilkan koneksi yang sedang aktif. *Server* basis data Borland Interbase (*ibserver.exe*) memiliki kondisi (*state*) LISTEN pada alamat IP lokal 0.0.0.0:3050. *Server* basis data Borland Interbase telah siap untuk menerima sebuah koneksi basis data pada *port* TCP 3050.

Process	PID	Local IP	Local Port	Remote IP	Remote Port	State	Protocol	Path
udp System	4	172.17.68.249	138			LISTEN	UDP	
udp System	4	172.17.68.249	137			LISTEN	UDP	
udp System	4	0.0.0.0	445			LISTEN	UDP	
tcp System	4	172.17.68.249	4575			LISTEN	TCP	
tcp System	4	172.17.68.249	139			LISTEN	TCP	
tcp System	4	0.0.0.0	445			LISTEN	TCP	
tcp cli.exe	436	127.0.0.1	1036	127.0.0.1	1030	ESTABLISHED	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
tcp cli.exe	436	127.0.0.1	1035	127.0.0.1	1030	ESTABLISHED	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
tcp cli.exe	436	127.0.0.1	1033	127.0.0.1	1030	ESTABLISHED	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
tcp cli.exe	436	127.0.0.1	1032			LISTEN	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
tcp cli.exe	436	127.0.0.1	1031			LISTEN	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
tcp cli.exe	436	127.0.0.1	1030	127.0.0.1	1036	ESTABLISHED	TCP	C:\Program Files\ATI Technologies\ATI ACE\cli.exe
udp lsass.exe	680	0.0.0.0	500			LISTEN	UDP	C:\WINDOWS\system32\lsass.exe
udp svchost.exe	868	0.0.0.0	135			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
tcp svchost.exe	868	0.0.0.0	135			LISTEN	TCP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	952	172.17.68.249	123			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	952	0.0.0.0	1026			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1048	0.0.0.0	4478			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1048	0.0.0.0	1064			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1048	0.0.0.0	1063			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1048	0.0.0.0	1062			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1048	0.0.0.0	1025			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
udp svchost.exe	1076	172.17.68.249	1300			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
tcp svchost.exe	1076	127.0.0.1	4606	127.0.0.1	1110	CLOSE_WAIT	TCP	C:\WINDOWS\system32\svchost.exe
tcp svchost.exe	1076	0.0.0.0	5000			LISTEN	TCP	C:\WINDOWS\system32\svchost.exe
tcp avp.exe	1300	0.0.0.0	1110			LISTEN	TCP	C:\Program Files\Kaspersky Lab\Kaspersky Anti-Virus 6.0\avp.exe
tcp ibserver.exe	1760	0.0.0.0	3050			LISTEN	TCP	C:\Program Files\Borland\InterBase\bin\ibserver.exe

**Gambar 6.8** Proses yang Berjalan pada Komputer *Server* Sebelum Ada Koneksi  
 Sumber: *Pengujian*

- Proses yang berjalan pada komputer *client* yang digunakan sebelum ada koneksi dengan komputer *server* ditunjukkan dalam Gambar 6.9. Perintah tersebut digunakan untuk menampilkan koneksi yang sedang aktif.

Process	PID	Local IP	Local Port	Remote IP	Remote Port	State	Protocol	Path
UDP System	4	172.17.68.242	138			LISTEN	UDP	
UDP System	4	172.17.68.242	137			LISTEN	UDP	
UDP System	4	0.0.0.0	445			LISTEN	UDP	
TCP System	4	172.17.68.242	1104			LISTEN	TCP	
TCP System	4	172.17.68.242	1085			LISTEN	TCP	
TCP System	4	172.17.68.242	1051			LISTEN	TCP	
TCP System	4	172.17.68.242	139			LISTEN	TCP	
TCP System	4	0.0.0.0	1034			LISTEN	TCP	
TCP System	4	0.0.0.0	445			LISTEN	TCP	
UDP lsass.exe	620	0.0.0.0	500			LISTEN	UDP	C:\WINDOWS\system32\lsass.exe
UDP svchost.exe	788	0.0.0.0	135			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
TCP svchost.exe	788	0.0.0.0	135			LISTEN	TCP	C:\WINDOWS\system32\svchost.exe
UDP svchost.exe	836	172.17.68.242	123			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
UDP svchost.exe	836	0.0.0.0	1027			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	836	0.0.0.0	3389			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	836	0.0.0.0	1025			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP mysqld-max.exe	892	0.0.0.0	3306			LISTEN	TCP	C:\mysql\bin\mysqld-max.exe
UDP svchost.exe	948	0.0.0.0	1026			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
UDP svchost.exe	1032	172.17.68.242	1900			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	1032	0.0.0.0	5000			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP Apache.exe	1348	0.0.0.0	80			LISTEN	TCP	C:\Apache\Apache2\bin\Apache.exe
TCP Unknown	1404	0.0.0.0	1028			LISTEN	TCP	

Gambar 6.9 Proses yang Berjalan pada Komputer *Client* Sebelum Ada Koneksi  
Sumber: Pengujian

- Proses yang berjalan pada komputer *client* yang digunakan setelah ada koneksi dengan komputer *server* ditunjukkan dalam Gambar 6.10. Koneksi terbentuk setelah komputer *client* membuka aplikasi Sistem Informasi Kehadiran Mahasiswa. Koneksi antara komputer *server* (alamat IP: 172.17.68.249) dengan komputer *client* (alamat IP: 172.17.68.242) ditunjukkan dengan adanya kondisi (*state*): ESTABLISHED. Kondisi (*state*) ESTABLISHED menunjukkan komputer *client* dapat terhubung dengan *server* basis data Borland Interbase di komputer *server*.

Process	PID	Local IP	Local Port	Remote IP	Remote Port	State	Protocol	Path
UDP System	4	172.17.68.242	138			LISTEN	UDP	
UDP System	4	172.17.68.242	137			LISTEN	UDP	
UDP System	4	0.0.0.0	445			LISTEN	UDP	
TCP System	4	172.17.68.242	1085	172.17.68.249	139	ESTABLISHED	TCP	
TCP System	4	172.17.68.242	1051			LISTEN	TCP	
TCP System	4	172.17.68.242	139			LISTEN	TCP	
TCP System	4	0.0.0.0	1034			LISTEN	TCP	
TCP System	4	0.0.0.0	445			LISTEN	TCP	
UDP lsass.exe	620	0.0.0.0	500			LISTEN	UDP	C:\WINDOWS\system32\lsass.exe
UDP svchost.exe	788	0.0.0.0	135			LISTEN	UDP	C:\WINDOWS\system32\svchost.exe
TCP svchost.exe	788	0.0.0.0	135			LISTEN	TCP	C:\WINDOWS\system32\svchost.exe
UDP svchost.exe	836	172.17.68.242	123			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
UDP svchost.exe	836	0.0.0.0	1027			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	836	0.0.0.0	3389			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	836	0.0.0.0	1025			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP mysqld-max.exe	892	0.0.0.0	3306			LISTEN	TCP	C:\mysql\bin\mysqld-max.exe
UDP svchost.exe	948	0.0.0.0	1026			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
UDP svchost.exe	1032	172.17.68.242	1900			LISTEN	UDP	C:\WINDOWS\System32\svchost.exe
TCP svchost.exe	1032	0.0.0.0	5000			LISTEN	TCP	C:\WINDOWS\System32\svchost.exe
TCP Apache.exe	1348	0.0.0.0	80			LISTEN	TCP	C:\Apache\Apache2\bin\Apache.exe
TCP Unknown	1404	0.0.0.0	1028			LISTEN	TCP	
TCP Client.exe	3836	172.17.68.242	1094	172.17.68.249	3050	ESTABLISHED	TCP	F:\data\itip\es Himura\gembu\Client.exe

Gambar 6.10 Proses yang Berjalan pada Komputer *Client* Setelah Ada Koneksi  
Sumber: Pengujian

- Koneksi *server* basis data Borland Interbase (*port* 3050) pada komputer *server* dapat dibangun menggunakan aplikasi sistem informasi kehadiran mahasiswa dari komputer *client*.

### 6.2.1.2 Pengujian Waktu Akses *Query*

#### A. Tujuan

Pengujian dilakukan untuk mengetahui *execution time query* pada basis data kehadiran.

Pengujian dilakukan untuk mendapatkan perbandingan waktu *query* yang dilakukan terhadap jumlah data yang berbeda pada basis data kehadiran.

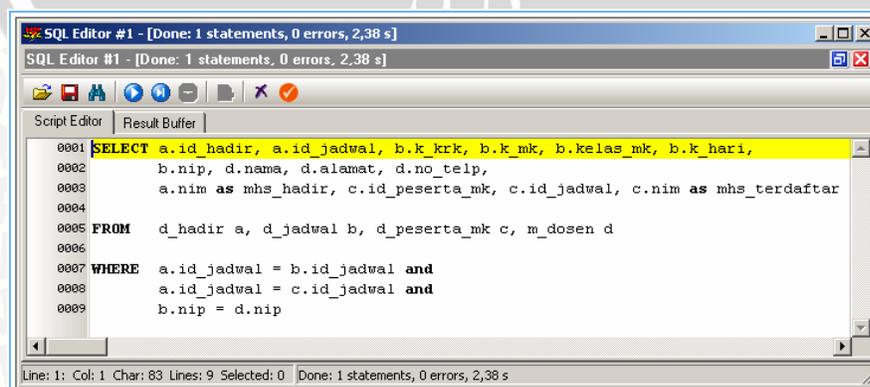
#### B. Prosedur Pengujian

PC *Client*:

- Proses pengujian waktu akses *query* dilakukan dengan memasukkan data linier pada tabel *d\_hadir*, *d\_jadwal*, *d\_peserta\_mk*, dan *m\_dosen* sebanyak 2000, 4000, 6000, 8000, dan 10000 data *entry*.
- Melakukan perintah `SELECT a.id_hadir, a.id_jadwal, b.k_krk, b.k_mk, b.kelas_mk, b.k_hari, b.nip, d.nama, d.alamat, d.no_telp, a.nim as mhs_hadir, c.id_peserta_mk, c.id_jadwal, c.nim as mhs_terdaftar FROM d_hadir a, d_jadwal b, d_peserta_mk c, m_dosen d WHERE a.id_jadwal = b.id_jadwal and a.id_jadwal = c.id_jadwal and b.nip = d.nip.`

#### C. Hasil Pengujian dan Analisis

- Proses pengujian waktu akses *query* pada basis data kehadiran ditunjukkan dalam Gambar 6.11.



```
SQL Editor #1 - [Done: 1 statements, 0 errors, 2,38 s]
SQL Editor #1 - [Done: 1 statements, 0 errors, 2,38 s]
Script Editor | Result Buffer
0001 SELECT a.id_hadir, a.id_jadwal, b.k_krk, b.k_mk, b.kelas_mk, b.k_hari,
0002 b.nip, d.nama, d.alamat, d.no_telp,
0003 a.nim as mhs_hadir, c.id_peserta_mk, c.id_jadwal, c.nim as mhs_terdaftar
0004
0005 FROM d_hadir a, d_jadwal b, d_peserta_mk c, m_dosen d
0006
0007 WHERE a.id_jadwal = b.id_jadwal and
0008 a.id_jadwal = c.id_jadwal and
0009 b.nip = d.nip
Line: 1 | Col: 1 | Char: 83 | Lines: 9 | Selected: 0 | Done: 1 statements, 0 errors, 2,38 s
```

Gambar 6.11 Proses pengujian waktu akses *query* terhadap basis data kehadiran  
Sumber: *Pengujian*

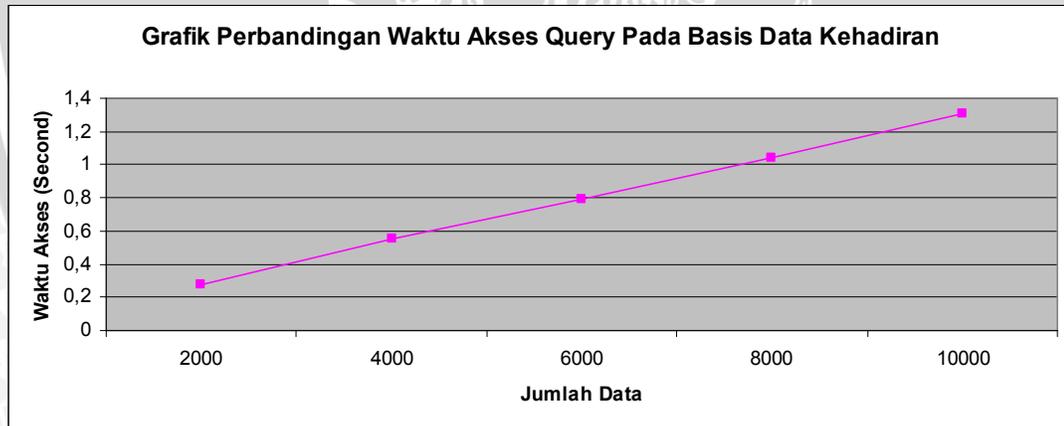
- Hasil rata-rata pengujian waktu akses *query* terhadap basis data kehadiran dengan jumlah data sebanyak 2000, 4000, 6000, 8000 dan 10000 data *entry* ditunjukkan dalam Tabel 6.1.

**Tabel 6.1** Tabel Rata-Rata Pengujian Waktu Akses *Query*

Tes	Jumlah data					Waktu akses (second)
	d_hadir	d_jadwal	d_peserta_mk	m_dosen	Total	
1	2000	2000	2000	2000	2000	0,28 s
2	4000	4000	4000	4000	4000	0,55 s
3	6000	6000	6000	6000	6000	0,79 s
4	8000	8000	8000	8000	8000	1,04 s
5	10000	10000	10000	10000	10000	1,31 s

Sumber: *Pengujian*

- Grafik perbandingan waktu akses *query* pada basis data kehadiran dengan jumlah data yang berbeda ditunjukkan dalam Gambar 6.12.



**Gambar 6.12** Grafik Perbandingan Waktu Akses *Query* pada Basis Data kehadiran  
Sumber: *Pengujian*

- Waktu akses *per-record* untuk satu *query* dapat diketahui dari hasil rata-rata pengujian waktu akses *query* yang ditunjukkan dalam Tabel 6.1. Waktu akses *per-record* merupakan hasil bagi antara rata-rata waktu akses dengan total *record*.

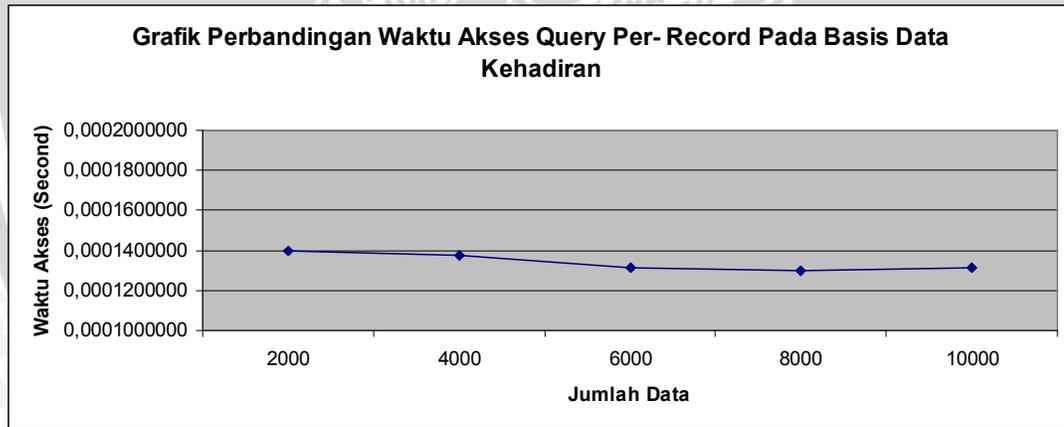
- Perbandingan rata-rata waktu akses *query* per-*record* pada basis data kehadiran ditunjukkan dalam Tabel 6.2.

**Tabel 6.2** Tabel rata-rata pengujian waktu akses *query* per-*record*

Tes	Total data	Waktu akses	Waktu akses per- <i>record</i>
1	2000	0,28 s	0,0001400 s
2	4000	0,55 s	0,0001375 s
3	6000	0,79 s	0,0001316 s
4	8000	1,04 s	0,0001300 s
5	10000	1,31 s	0,0001310 s
Rata-rata Waktu Akses <i>Query</i> Per- <i>Record</i>			0,0001340 s

Sumber: *Pengujian*

- Nilai rata-rata waktu akses *query* per-*record* pada basis data kehadiran yang ditunjukkan dalam Tabel 6.2 adalah 0,0001340 *second*.
- Grafik perbandingan waktu akses *query* per-*record* pada basis data kehadiran ditunjukkan dalam Gambar 6.13.



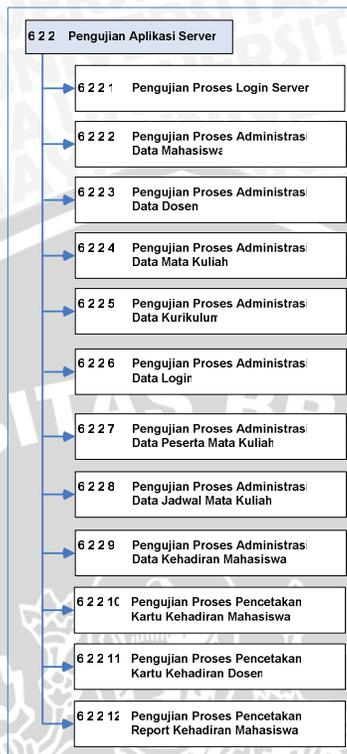
**Gambar 6.13** Grafik Perbandingan Waktu Akses *Query* Per-*Record* pada Basis Data kehadiran

Sumber: *Pengujian*

### 6.2.2 Pengujian Aplikasi *Server*

Pengujian aplikasi *server* secara garis besar meliputi pengujian proses administrasi data dan proses pencetakan kartu. Pengujian proses administrasi data terdiri atas penambahan data, pengubahan data, penghapusan data, dan pencarian data. Pengujian proses pencetakan kartu dilakukan untuk menguji pencetakan kartu sesuai dengan masukan data pada *form* pencetakan kartu.

Diagram pohon pengujian aplikasi *server* ditunjukkan dalam Gambar 6.14.



Gambar 6.14 Diagram Pohon Pengujian Aplikasi *Server*  
Sumber: *Perancangan*

### 6.2.2.1 Pengujian Proses *Login Server*

#### A. Tujuan

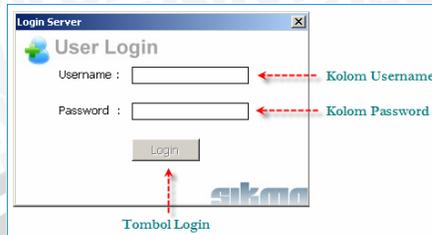
- Pengujian dilakukan untuk mengetahui apakah administrator dapat melakukan proses *login* pada aplikasi *server*.

#### B. Prosedur Pengujian

- *Window command prompt* dijalankan dengan memberikan perintah sebagai berikut:  
Start | Run... | Open: cmd.exe
- Aplikasi *server* dijalankan dengan memanggil:  
G:\! Delphi\delphi\_5\server.exe
- Melakukan proses *login* pada aplikasi *server* dengan *username* admin dan *password* admin.
- Melakukan proses *logout* pada aplikasi *server*.

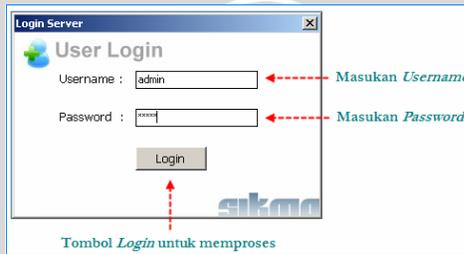
**C. Hasil Pengujian dan Analisis**

- Halaman antarmuka aplikasi *server* untuk proses *login* ditunjukkan dalam Gambar 6.15.



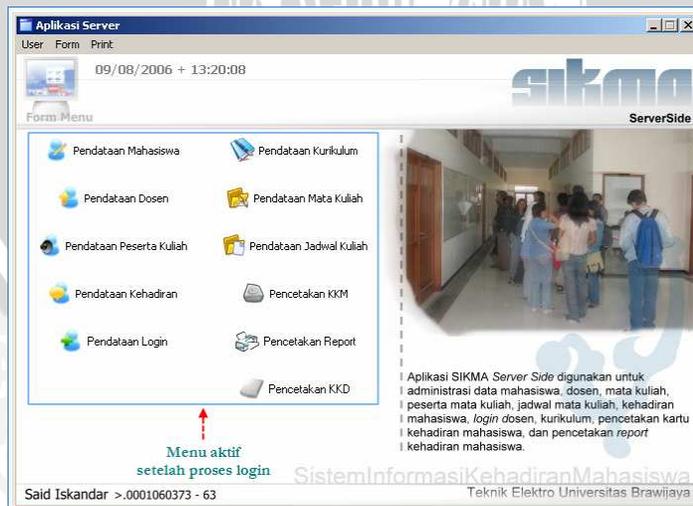
**Gambar 6.15** Halaman Antarmuka Proses *Login*  
Sumber: *Pengujian*

- Proses *login* yang dilakukan oleh administrator ditunjukkan dalam Gambar 6.16. Proses *login* menggunakan *username* admin dan *password* admin.



**Gambar 6.16** Proses *Login* Administrator  
Sumber: *Pengujian*

- Proses *login* yang berhasil dilakukan oleh administrator ditunjukkan dalam Gambar 6.17. Halaman utama aplikasi *server* akan ditampilkan setelah proses *login* selesai dilakukan.



**Gambar 6.17** Menu Aktif setelah Proses *Login*  
Sumber: *Pengujian*

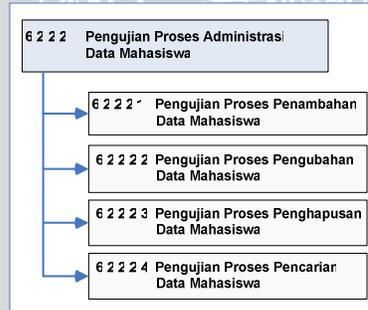
- Proses *logout* pada aplikasi *server* dilakukan apabila administrator ingin menutup aplikasi *server*. Proses *logout* yang berhasil dilakukan oleh administrator ditunjukkan dalam Gambar 6.18.



**Gambar 6.18** Menu Tidak Aktif setelah Proses *Logout*  
Sumber: *Pengujian*

### 6.2.2.2 Pengujian Proses Administrasi Data Mahasiswa

Pengujian proses administrasi data mahasiswa meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data mahasiswa ditunjukkan dalam Gambar 6.19.



**Gambar 6.19** Diagram Pohon Pengujian Proses Administrasi Data Mahasiswa  
Sumber: *Perancangan*

#### 6.2.2.2.1 Pengujian Proses Penambahan Data Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mahasiswa dapat digunakan untuk menambah data mahasiswa yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mahasiswa pada halaman utama aplikasi *server*.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data mahasiswa baru dan memasukkan data mahasiswa baru ke dalam tabel data mahasiswa.
- Administrator menyimpan data mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

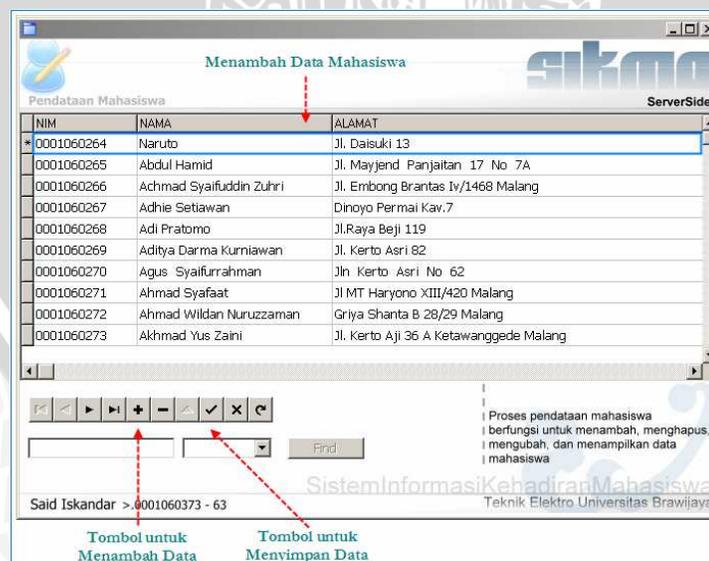
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data mahasiswa baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mhs ORDER BY nim rows 10;
```

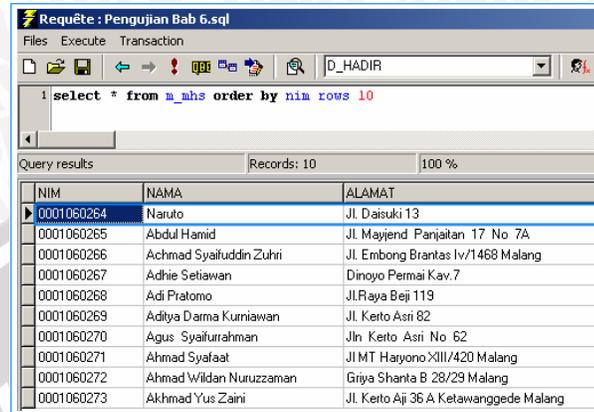
## C. Hasil Pengujian dan Analisis

- Administrator mengisi tabel data mahasiswa dengan masukan data mahasiswa baru yang terdiri atas nim, nama, dan alamat. Administrator mengisi *record* kosong dengan NIM '0001060264', NAMA 'Naruto', dan ALAMAT 'Jl. Daisuki 13'. Tampilan antarmuka proses penambahan data mahasiswa ditunjukkan dalam Gambar 6.20.



**Gambar 6.20** Antarmuka Proses Penambahan Data Mahasiswa  
Sumber: *Pengujian*

- Data mahasiswa yang telah ditambahkan akan disimpan pada tabel `m_mhs`. Hasil proses penambahan data mahasiswa baru pada basis data kehadiran ditunjukkan dalam Gambar 6.21.



NIM	NAMA	ALAMAT
0001060264	Naruto	Jl. Daisuki 13
0001060265	Abdul Hamid	Jl. Mayjend Panjaitan 17 No 7A
0001060266	Achmad Syaifuddin Zuhri	Jl. Embong Brantas Iv/1468 Malang
0001060267	Adhie Setiawan	Dinoyo Permai Kav.7
0001060268	Adi Pratomo	Jl. Raya Beji 119
0001060269	Aditya Darma Kurniawan	Jl. Kerto Astri 82
0001060270	Agus Syaifurrahman	Jln Kerto Astri No 62
0001060271	Ahmad Syafaat	Jl MT Haryono XIII/420 Malang
0001060272	Ahmad Wildan Nuruzzaman	Griya Shantia B 28/29 Malang
0001060273	Akhmed Yus Zaini	Jl. Kerto Aji 36 A Ketawanggede Malang

**Gambar 6.21** Hasil Penambahan Data Mahasiswa pada Basis Data Kehadiran  
Sumber: *Pengujian*

#### 6.2.2.2.2 Pengujian Proses Perubahan Data Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mahasiswa dapat digunakan untuk mengubah data mahasiswa yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mahasiswa pada halaman utama aplikasi *server*.
- Administrator mengubah data mahasiswa dengan menekan tombol navigasi bertanda (▲) dan memasukkan data mahasiswa yang akan diubah ke dalam tabel data mahasiswa.
- Administrator menyimpan data mahasiswa pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

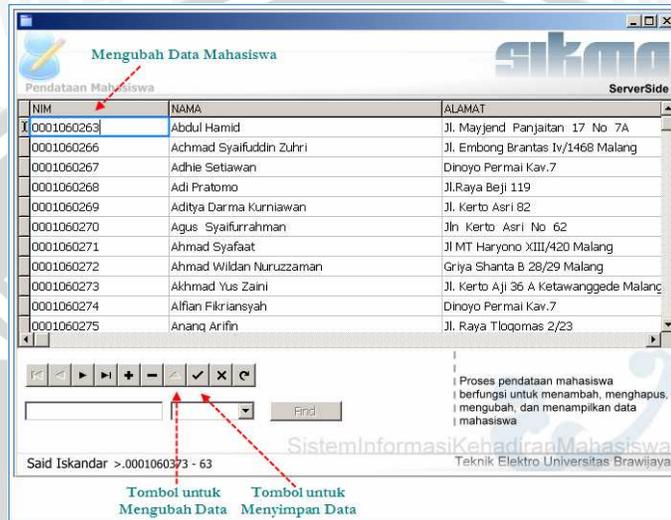
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data mahasiswa yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mhs ORDER BY nim rows 10;
```

**C. Hasil Pengujian dan Analisis**

- Administrator mengubah isi tabel data mahasiswa dengan masukan data mahasiswa berupa nim. Administrator mengganti NIM ‘0001060265’ dengan NIM ‘0001060263’. Tampilan antarmuka proses pengubahan data mahasiswa ditunjukkan dalam Gambar 6.22.



**Gambar 6.22** Antarmuka Proses Pengubahan Data Mahasiswa  
Sumber: *Pengujian*

- Data mahasiswa yang telah diubah akan disimpan pada tabel m\_mhs. Hasil proses pengubahan data mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.23.



**Gambar 6.23** Hasil Pengubahan Data Mahasiswa pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.2.3 Pengujian Proses Penghapusan Data Mahasiswa

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mahasiswa dapat digunakan untuk menghapus data mahasiswa yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mahasiswa pada halaman utama aplikasi server.
- Administrator dapat menghapus data mahasiswa dengan menekan tombol navigasi bertanda (–) setelah memilih data mahasiswa yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel m\_mhs.

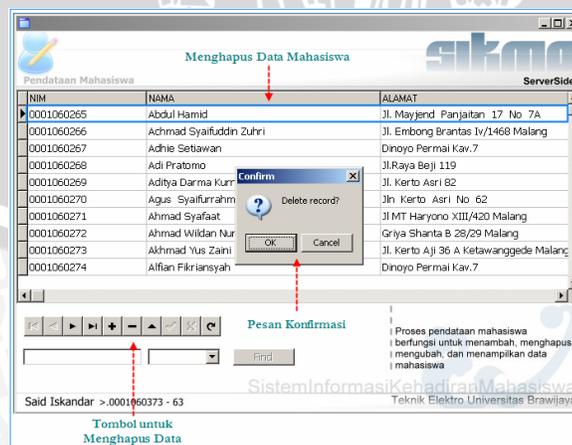
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data mahasiswa sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:  

```
SELECT * FROM m_mhs ORDER BY nim rows 10;
```

#### C. Hasil Pengujian dan Analisis

- Administrator menghapus *record* data dengan NIM '0001060265', NAMA 'Abdul Hamid', dan ALAMAT 'Jl. Mayjend Panjaitan 17 No. 7A'. Tampilan antarmuka proses penghapusan data mahasiswa ditunjukkan dalam Gambar 6.24.



Gambar 6.24 Antarmuka Proses Penghapusan Data Mahasiswa  
Sumber: Pengujian

- Data mahasiswa sebelum proses penghapusan data mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.25.

The screenshot shows a query window titled 'Requête : Pengujian Bab 6.sql'. The query is 'select \* from m\_mhs order by nim rows 10'. The results table has 10 records with columns NIM, NAMA, and ALAMAT.

NIM	NAMA	ALAMAT
0001060263	Abdul Hamid	Jl. Mayjend Panjaitan 17 No 7A
0001060264	Naruto	Jl. Daisuki 13
0001060266	Achmad Syaifuddin Zuhri	Jl. Embong Brantas Iv/1468 Malang
0001060267	Adhrie Setiawan	Dinoyo Permai Kav.7
0001060268	Adi Pratomo	Jl.Raya Beji 119
0001060269	Aditya Darma Kurniawan	Jl. Kerto Asri 82
0001060270	Agus Syaifurrahman	Jln Kerto Asri No 62
0001060271	Ahmad Syafaat	Jl MT Haryono XIII/420 Malang
0001060272	Ahmad Wildan Nuruzzaman	Griya Shanta B 28/29 Malang
0001060273	Akhmad Yus Zaini	Jl. Kerto Aji 36 A Ketawanggede Malang

**Gambar 6.25** Data Mahasiswa sebelum Proses Penghapusan Data Mahasiswa  
Sumber: *Pengujian*

- Data mahasiswa setelah proses penghapusan data mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.26.

The screenshot shows a query window titled 'Query: New Script'. The query is 'select \* from m\_mhs order by nim rows 10'. The results table has 10 records with columns NIM, NAMA, and ALAMAT.

NIM	NAMA	ALAMAT
0001060266	Achmad Syaifuddin Zuhri	Jl. Embong Brantas Iv/1468 Malang
0001060267	Adhrie Setiawan	Dinoyo Permai Kav.7
0001060268	Adi Pratomo	Jl.Raya Beji 119
0001060269	Aditya Darma Kurniawan	Jl. Kerto Asri 82
0001060270	Agus Syaifurrahman	Jln Kerto Asri No 62
0001060271	Ahmad Syafaat	Jl MT Haryono XIII/420 Malang
0001060272	Ahmad Wildan Nuruzzaman	Griya Shanta B 28/29 Malang
0001060273	Akhmad Yus Zaini	Jl. Kerto Aji 36 A Ketawanggede Malang
0001060274	Allian Fikriansyah	Dinoyo Permai Kav.7
0001060275	Anang Aritin	Jl. Raya Tlogomas 2/23

**Gambar 6.26** Data Mahasiswa setelah Proses Penghapusan Data Mahasiswa  
Sumber: *Pengujian*

#### 6.2.2.2.4 Pengujian Proses Pencarian Data Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mahasiswa dapat digunakan untuk mencari data mahasiswa yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.

- Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian.

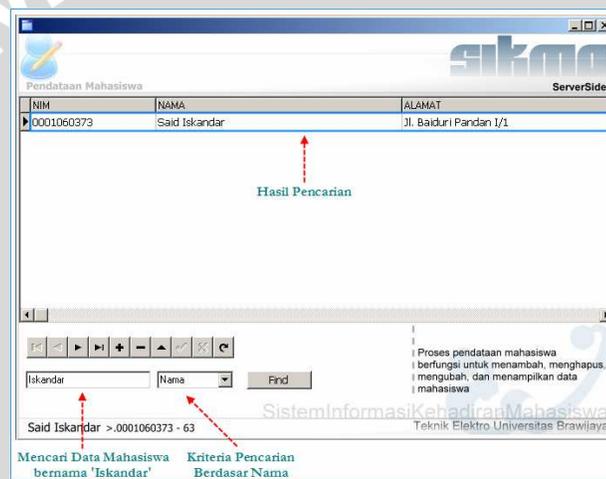
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat hasil pencarian data mahasiswa dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mhs WHERE nama CONTAINING 'iskandar';
```

### C. Hasil Pengujian dan Analisis

- Administrator memilih kriteria pencarian berupa Nama dan mengisi kata kunci pencarian dengan Nama 'Iskandar' pada masukan pencarian. Tampilan antarmuka proses pencarian data mahasiswa ditunjukkan dalam Gambar 6.27.



**Gambar 6.27** Antarmuka Proses Pencarian Data Mahasiswa  
Sumber: *Pengujian*

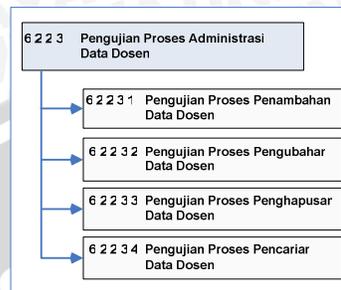
- Hasil proses pencarian menunjukkan bahwa data mahasiswa dengan Nama 'Iskandar' telah tersimpan pada basis data kehadiran. Hasil proses pencarian data mahasiswa ditunjukkan dalam Gambar 6.28.



**Gambar 6.28** Hasil Proses Pencarian Data Mahasiswa pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.3 Pengujian Proses Administrasi Data Dosen

Pengujian proses administrasi data dosen meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data dosen ditunjukkan dalam Gambar 6.29.



**Gambar 6.29** Diagram Pohon Pengujian Proses Administrasi Data Dosen  
Sumber: *Perancangan*

#### 6.2.2.3.1 Pengujian Proses Penambahan Data Dosen

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data dosen dapat digunakan untuk menambah data dosen yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

###### PC Client:

- Administrator memilih menu Pendataan Dosen pada halaman utama aplikasi server.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data dosen baru dan memasukkan data dosen baru ke dalam tabel data dosen.
- Administrator menyimpan data dosen baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

###### PC Server:

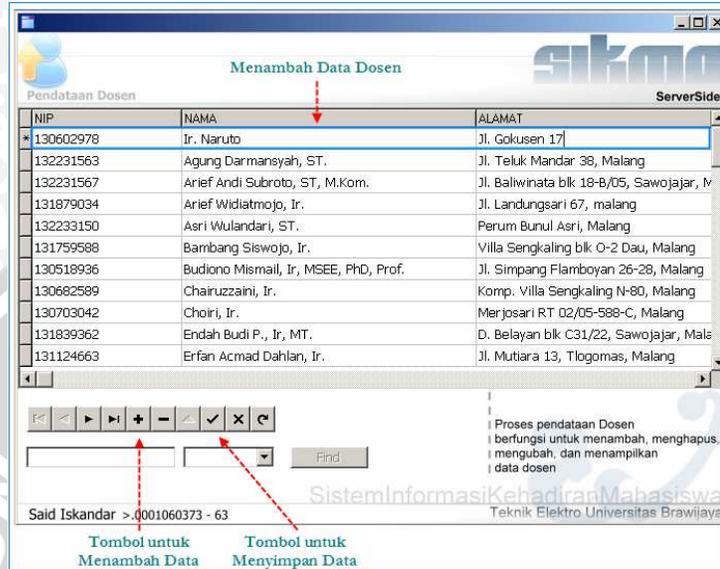
- Server basis data kehadiran dijalankan untuk melihat data dosen baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_dosen ORDER BY nip ROWS 5;
```

##### C. Hasil Pengujian dan Analisis

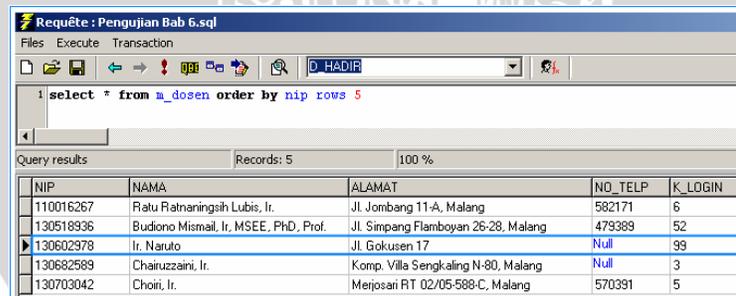
- Administrator mengisi tabel data dosen dengan masukan data dosen baru yang terdiri atas nip, nama, alamat, dan kode *login*. Administrator mengisi *record*

kosong dengan NIP '130 602 978', NAMA 'Ir. Naruto', ALAMAT 'Jl. Gokusen 17', dan k\_login '99'. Tampilan antarmuka proses penambahan data dosen ditunjukkan dalam Gambar 6.30.



Gambar 6.30 Antarmuka Proses Penambahan Data Dosen  
Sumber: Pengujian

- Data dosen yang telah ditambahkan akan disimpan pada tabel m\_dosen. Hasil proses penambahan data dosen baru pada basis data kehadiran ditunjukkan dalam Gambar 6.31.



Gambar 6.31 Hasil Penambahan Data Dosen pada Basis Data Kehadiran  
Sumber: Pengujian

### 6.2.2.3.2 Pengujian Proses Pengubahan Data Dosen

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data dosen dapat digunakan untuk mengubah data dosen yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Dosen pada halaman utama aplikasi server.
- Administrator mengubah data dosen dengan menekan tombol navigasi bertanda (▲) dan memasukkan data dosen yang akan diubah ke dalam tabel data dosen.
- Administrator menyimpan data dosen pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

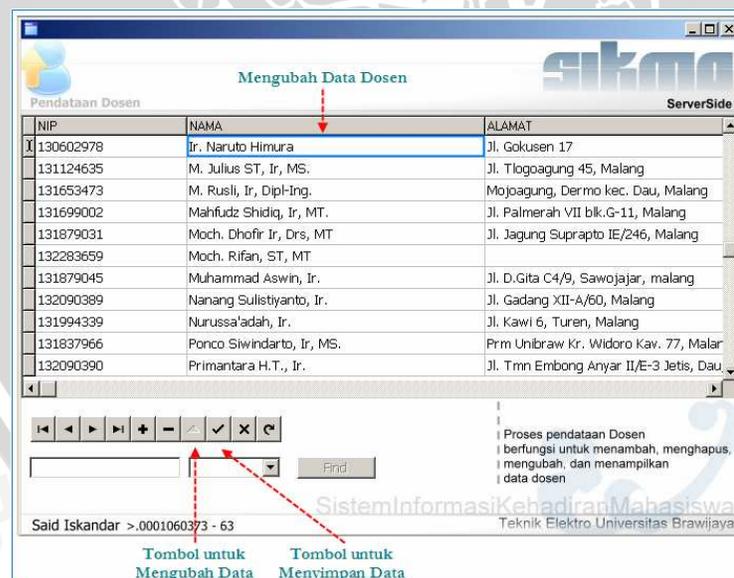
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data dosen yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_dosen ORDER BY nip ROWS 5;
```

## C. Hasil Pengujian dan Analisis

- Administrator mengubah isi tabel data dosen dengan masukan data dosen berupa nip. Administrator mengganti NAMA 'Ir. Naruto' dengan NAMA 'Ir. Naruto Himura'. Tampilan antarmuka proses pengubahan data dosen ditunjukkan dalam Gambar 6.32.



**Gambar 6.32** Antarmuka Proses Pengubahan Data Dosen  
Sumber: *Pengujian*

- Data dosen yang telah diubah akan disimpan pada tabel `m_dosen`. Hasil proses pengubahan data dosen pada basis data kehadiran ditunjukkan dalam Gambar 6.33.

Requête : Pengujian Bab 6.sql  
Files Execute Transaction  
D\_HADIR

```
1 select * from m_dosen order by nip rows 5
```

Query results Records: 5 100 %

NIP	NAMA	ALAMAT	NO_TELP	K_LOGIN
110016267	Ratu Ratnaningsih Lubis, Ir.	Jl. Jombang 11-A, Malang	582171	6
130518936	Budiono Mismail, Ir, MSEE, PhD, Prof.	Jl. Simpang Flamboyan 26-28, Malang	479389	52
130602978	Ir. Naruto Himura	Jl. Gokusen 17	Null	99
130682589	Chairuzzaini, Ir.	Komp. Villa Sengkaling N-80, Malang	Null	3
130703042	Choiiri, Ir.	Merjosari RT 02/05-588-C, Malang	570391	5

**Gambar 6.33** Hasil Pengubahan Data Dosen pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.3.3 Pengujian Proses Penghapusan Data Dosen

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data dosen dapat digunakan untuk menghapus data dosen yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

##### PC Client:

- Administrator memilih menu Pendataan Dosen pada halaman utama aplikasi server.
- Administrator dapat menghapus data dosen dengan menekan tombol navigasi bertanda (-) setelah memilih data dosen yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `m_dosen`.

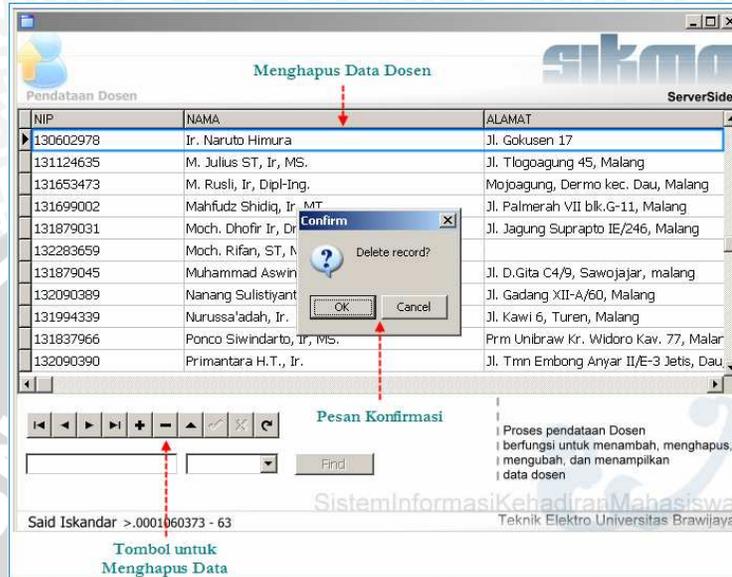
##### PC Server:

- Server basis data kehadiran dijalankan untuk melihat data dosen sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_dosen ORDER BY nip ROWS 5;
```

**C. Hasil Pengujian dan Analisis**

- Administrator akan menghapus *record* data dengan NIP ‘130 602 978’, NAMA ‘Ir. Naruto Himura’, dan ALAMAT ‘Jl. Gokusen 17’. Tampilan antarmuka proses penghapusan data dosen ditunjukkan dalam Gambar 6.34.



**Gambar 6.34** Antarmuka Proses Penghapusan Data Dosen  
Sumber: *Pengujian*

- Data dosen sebelum proses penghapusan data dosen pada basis data kehadiran ditunjukkan dalam Gambar 6.35.



**Gambar 6.35** Data Dosen sebelum Proses Penghapusan Data Dosen  
Sumber: *Pengujian*

- Data dosen setelah proses penghapusan data dosen pada basis data kehadiran ditunjukkan dalam Gambar 6.36.



**Gambar 6.36** Data Dosen setelah Proses Penghapusan Data Dosen  
Sumber: *Pengujian*

#### 6.2.2.3.4 Pengujian Proses Pencarian Data Dosen

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data dosen dapat digunakan untuk mencari data dosen yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.
- Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian.

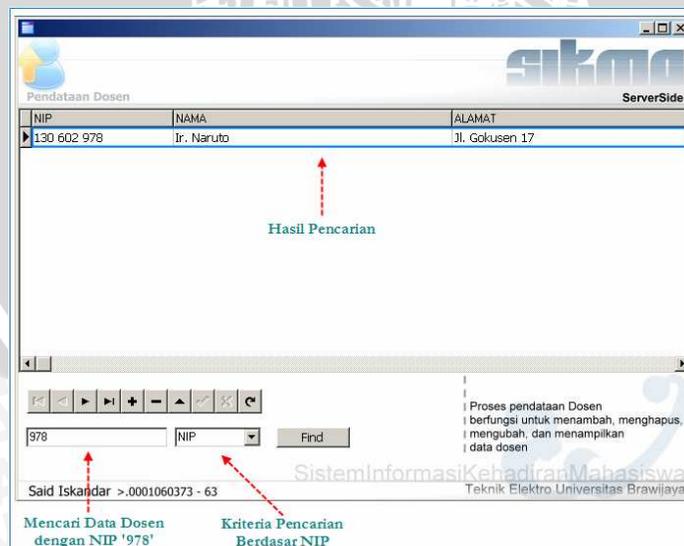
PC Server:

- Server basis data kehadiran dijalankan untuk melihat hasil pencarian data dosen dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_dosen WHERE nip CONTAINING '978';
```

##### C. Hasil Pengujian dan Analisis

- Administrator memilih kriteria pencarian berupa NIP dan mengisi kata kunci pencarian dengan NIP '978' pada masukan pencarian. Tampilan antarmuka proses pencarian data dosen ditunjukkan dalam Gambar 6.37.



Gambar 6.37 Antarmuka Proses Pencarian Data Dosen  
Sumber: Pengujian

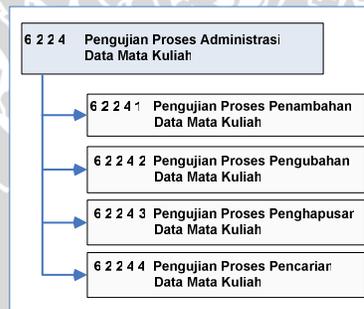
- Hasil proses pencarian menunjukkan bahwa data dosen dengan nip mengandung angka ‘978’ telah tersimpan pada basis data kehadiran. Hasil proses pencarian data dosen ditunjukkan dalam Gambar 6.38.



Gambar 6.38 Hasil Proses Pencarian Data Dosen pada Basis Data Kehadiran  
Sumber: *Pengujian*

#### 6.2.2.4 Pengujian Proses Administrasi Data Mata Kuliah

Pengujian proses administrasi data mata kuliah meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data mata kuliah ditunjukkan dalam Gambar 6.39.



Gambar 6.39 Diagram Pohon Pengujian Proses Administrasi Data Mata Kuliah  
Sumber: *Perancangan*

##### 6.2.2.4.1 Pengujian Proses Penambahan Data Mata Kuliah

###### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mata kuliah dapat digunakan untuk menambah data mata kuliah yang terdapat pada basis data kehadiran.

###### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mata Kuliah pada halaman utama aplikasi server.

- Administrator menekan tombol navigasi bertanda (+) untuk menambah data mata kuliah baru dan memasukkan data mata kuliah baru ke dalam tabel data mata kuliah.
- Administrator menyimpan data mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

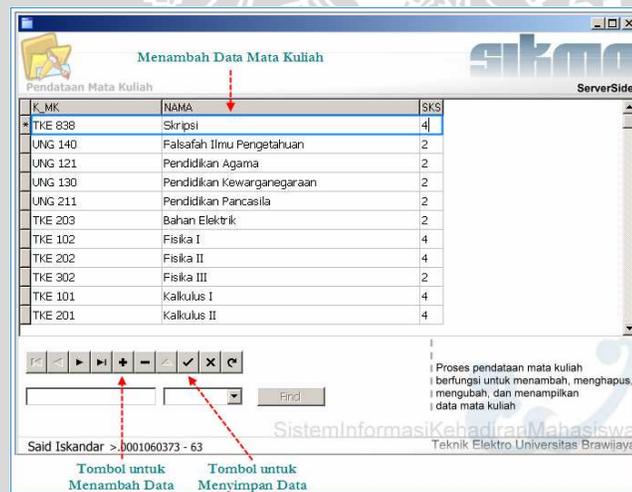
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data mata kuliah baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mk ORDER BY k_mk DESC ROWS 6;
```

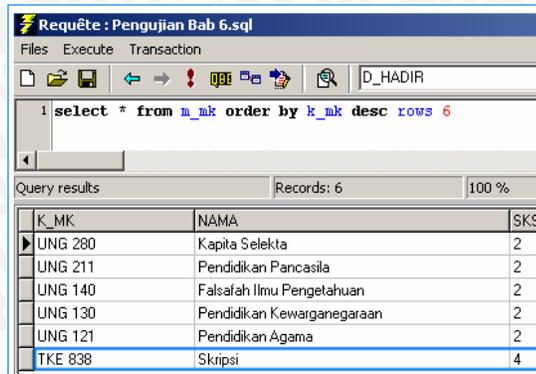
### C. Hasil Pengujian dan Analisis

- Administrator mengisi tabel data mata kuliah dengan masukan data mata kuliah baru yang terdiri atas kode mata kuliah, nama, dan sks. Administrator mengisi *record* kosong dengan K\_MK 'TKE 838', NAMA 'Skripsi', dan SKS '4'. Tampilan antarmuka proses penambahan data mata kuliah ditunjukkan dalam Gambar 6.40.



**Gambar 6.40** Antarmuka Proses Penambahan Data Mata Kuliah  
Sumber: *Pengujian*

- Data mata kuliah yang telah ditambahkan akan disimpan pada tabel *m\_mk*. Hasil proses penambahan data mata kuliah baru pada basis data kehadiran ditunjukkan dalam Gambar 6.41.



K_MK	NAMA	SKS
UNG 280	Kapita Selektta	2
UNG 211	Pendidikan Pancasila	2
UNG 140	Falsafah Ilmu Pengetahuan	2
UNG 130	Pendidikan Kewarganegaraan	2
UNG 121	Pendidikan Agama	2
TKE 838	Skripsi	4

Gambar 6.41 Hasil Penambahan Data Mata Kuliah pada Basis Data Kehadiran  
Sumber: *Pengujian*

#### 6.2.2.4.2 Pengujian Proses Perubahan Data Mata Kuliah

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mata kuliah dapat digunakan untuk mengubah data mata kuliah yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mata Kuliah pada halaman utama aplikasi server.
- Administrator mengubah data mata kuliah dengan menekan tombol navigasi bertanda (▲) dan memasukkan data mata kuliah yang akan diubah ke dalam tabel data mata kuliah.
- Administrator menyimpan data mata kuliah pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

PC Server:

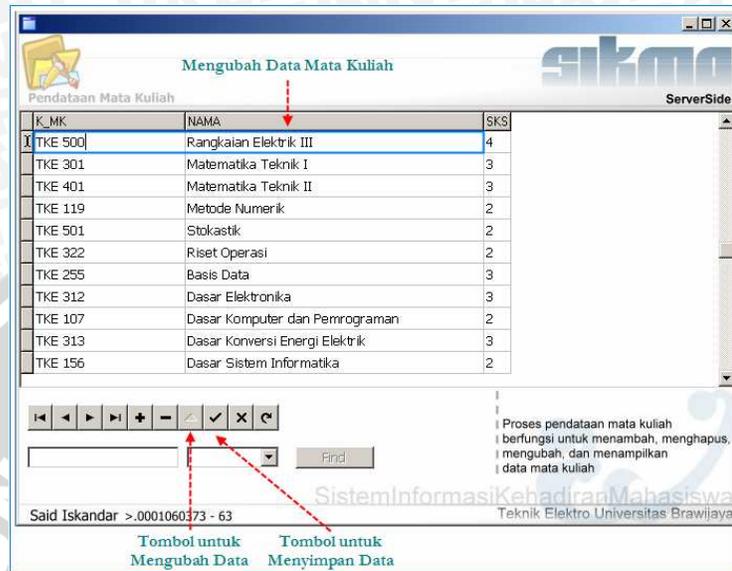
- Server basis data kehadiran dijalankan untuk melihat data mata kuliah yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mk WHERE k_mk BETWEEN 'TKE 489' and 'TKE 506' ORDER BY k_mk;
```

##### C. Hasil Pengujian dan Analisis

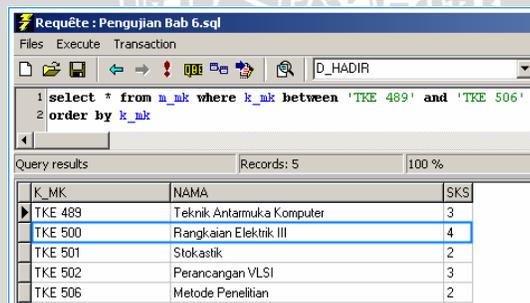
- Administrator mengubah isi tabel data mata kuliah dengan masukan data mata kuliah berupa kode mata kuliah dan nama mata kuliah. Administrator mengganti

K\_MK 'TKE 260' dengan K\_MK 'TKE 500' dan NAMA 'Jaringan Komputer' dengan NAMA 'Rangkaian Elektrik III'. Tampilan antarmuka proses pengubahan data mata kuliah ditunjukkan dalam Gambar 6.42.



Gambar 6.42 Antarmuka Proses Pengubahan Data Mata Kuliah  
Sumber: Pengujian

- Data mata kuliah yang telah diubah akan disimpan pada tabel m\_mk. Hasil proses pengubahan data mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.43.



Gambar 6.43 Hasil Pengubahan Data Mata Kuliah pada Basis Data Kehadiran  
Sumber: Pengujian

### 6.2.2.4.3 Pengujian Proses Penghapusan Data Mata Kuliah

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mata kuliah dapat digunakan untuk menghapus data mata kuliah yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Mata Kuliah pada halaman utama aplikasi *server*.
- Administrator dapat menghapus data mata kuliah dengan menekan tombol navigasi bertanda (–) setelah memilih data mata kuliah yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel *m\_mk*.

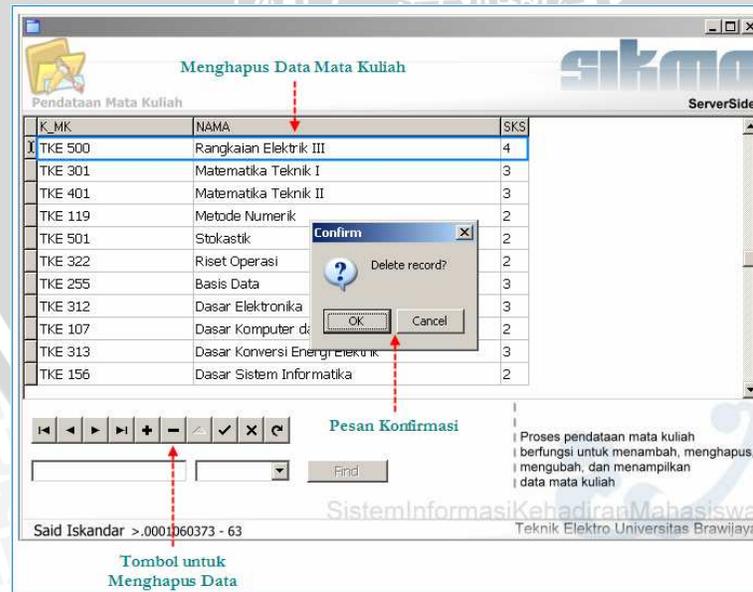
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data dosen sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:  

```
SELECT * FROM m_mk WHERE k_mk BETWEEN 'TKE 489' and 'TKE 506' ORDER BY k_mk;
```

## C. Hasil Pengujian dan Analisis

- Administrator akan menghapus *record* data dengan K\_MK 'TKE 500', NAMA 'Rangkaian Elektrik III', dan SKS '4'. Tampilan antarmuka proses penghapusan data mata kuliah ditunjukkan dalam Gambar 6.44.



**Gambar 6.44** Antarmuka Proses Penghapusan Data Mata Kuliah  
 Sumber: Pengujian

- Data mata kuliah sebelum proses penghapusan data mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.45.

K_MK	NAMA	SKS
TKE 489	Teknik Antarmuka Komputer	3
TKE 500	Rangkaian Elektrik III	4
TKE 501	Stokastik	2
TKE 502	Perancangan VLSI	3
TKE 506	Metode Penelitian	2

**Gambar 6.45** Data Mata Kuliah sebelum Proses Penghapusan Data Mata Kuliah  
Sumber: *Pengujian*

- Data mata kuliah setelah proses penghapusan data mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.46.

K_MK	NAMA	SKS
TKE 489	Teknik Antarmuka Komputer	3
TKE 501	Stokastik	2
TKE 502	Perancangan VLSI	3
TKE 506	Metode Penelitian	2

**Gambar 6.46** Data Mata Kuliah setelah Proses Penghapusan Data Mata Kuliah  
Sumber: *Pengujian*

#### 6.2.2.4.4 Pengujian Proses Pencarian Data Mata Kuliah

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data mata kuliah dapat digunakan untuk mencari data mata kuliah yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

*PC Client:*

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.
- Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian.

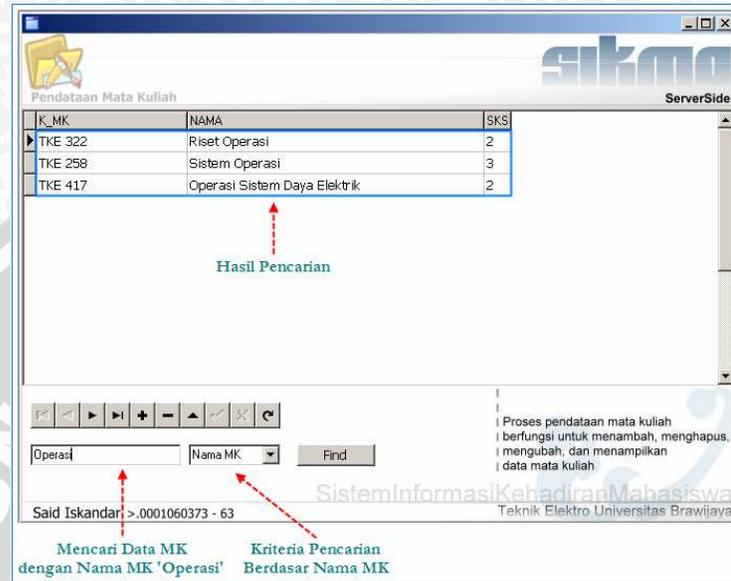
*PC Server:*

- *Server* basis data kehadiran dijalankan untuk melihat hasil pencarian data mata kuliah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_mk WHERE nama CONTAINING 'Operasi';
```

### C. Hasil Pengujian dan Analisis

- Administrator memilih kriteria pencarian berupa NAMA dan mengisi kata kunci pencarian dengan NAMA 'Operasi' pada masukan pencarian. Tampilan antarmuka proses pencarian data mata kuliah ditunjukkan dalam Gambar 6.47.



**Gambar 6.47** Antarmuka Proses Pencarian Data Mata Kuliah  
Sumber: *Pengujian*

- Hasil proses pencarian menunjukkan bahwa data mata kuliah dengan nama mengandung 'Operasi' telah tersimpan pada basis data kehadiran. Hasil proses pencarian data mata kuliah ditunjukkan dalam Gambar 6.48.

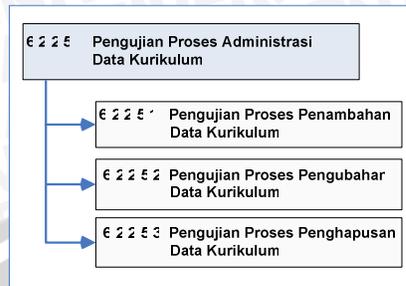


**Gambar 6.48** Hasil Proses Pencarian Data Mata Kuliah pada Basis Data Kehadiran  
Sumber: *Pengujian*

#### 6.2.2.5 Pengujian Proses Administrasi Data Kurikulum

Pengujian proses administrasi data kurikulum meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data.

Diagram pohon pengujian proses administrasi data kurikulum ditunjukkan dalam Gambar 6.49.



**Gambar 6.49** Diagram Pohon Pengujian Proses Administrasi Data Kurikulum  
Sumber: *Perancangan*

### 6.2.2.5.1 Pengujian Proses Penambahan Data Kurikulum

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kurikulum dapat digunakan untuk menambah data kurikulum yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Kurikulum pada halaman utama aplikasi *server*.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data kurikulum baru dan memasukkan data kurikulum baru ke dalam tabel data kurikulum.
- Administrator menyimpan data kurikulum baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

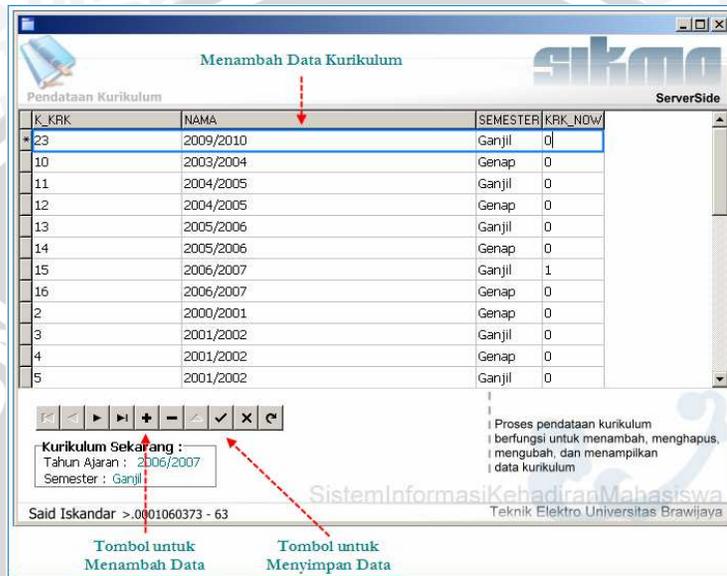
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data kurikulum baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_krk ORDER BY k_krk ROWS 10;
```

**C. Hasil Pengujian dan Analisis**

- Administrator mengisi tabel data kurikulum dengan masukan data kurikulum baru yang terdiri atas kode kurikulum, nama, semester, dan kurikulum yang aktif. Administrator mengisi *record* kosong dengan K\_KRK '23', NAMA '2009/2010', SEMESTER 'Ganjil' dan KRK\_NOW '0'. Tampilan antarmuka proses penambahan data kurikulum ditunjukkan dalam Gambar 6.50.



**Gambar 6.50** Antarmuka Proses Penambahan Data Kurikulum  
Sumber: *Pengujian*

- Data kurikulum yang telah ditambahkan akan disimpan pada tabel *m\_krk*. Hasil proses penambahan data kurikulum baru pada basis data kehadiran ditunjukkan dalam Gambar 6.51.



**Gambar 6.51** Hasil Penambahan Data Kurikulum pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.5.2 Pengujian Proses Pengubahan Data Kurikulum

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kurikulum dapat digunakan untuk mengubah data kurikulum yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Kurikulum pada halaman utama aplikasi server.
- Administrator mengubah data kurikulum dengan menekan tombol navigasi bertanda (▲) dan memasukkan data kurikulum yang akan diubah ke dalam tabel data kurikulum.
- Administrator menyimpan data kurikulum pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

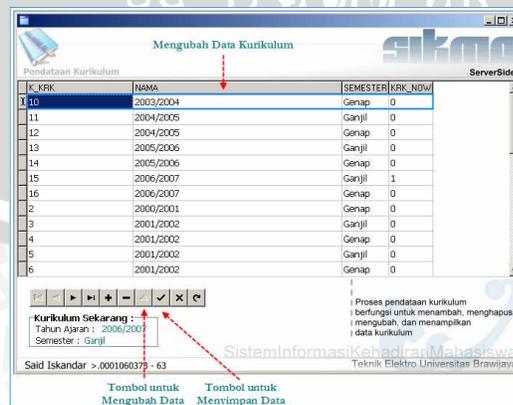
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data kurikulum yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_krk ORDER BY k_krk ROWS 5;
```

#### C. Hasil Pengujian dan Analisis

- Administrator mengubah isi tabel data kurikulum dengan masukan data kurikulum berupa nama. Administrator mengganti NAMA '2003/2004' dengan NAMA '2020/2021'. Tampilan antarmuka proses pengubahan data kurikulum ditunjukkan dalam Gambar 6.52.



**Gambar 6.52** Antarmuka Proses Pengubahan Data Kurikulum

Sumber: *Pengujian*

- Data kurikulum yang telah diubah akan disimpan pada tabel `m_krk`. Hasil proses perubahan data kurikulum pada basis data kehadiran ditunjukkan dalam Gambar 6.53.



K_KRK	NAMA	SEMESTER	KRK_NOW
1	2000/2001	Ganjil	0
10	2020/2021	Genap	0
11	2004/2005	Ganjil	0
12	2004/2005	Genap	0
13	2005/2006	Ganjil	0

Gambar 6.53 Hasil Pengubahan Data Kurikulum pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.5.3 Pengujian Proses Penghapusan Data Kurikulum

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kurikulum dapat digunakan untuk menghapus data kurikulum yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

##### PC Client:

- Administrator memilih menu Pendataan Kurikulum pada halaman utama aplikasi *server*.
- Administrator dapat menghapus data kurikulum dengan menekan tombol navigasi bertanda (–) setelah memilih data kurikulum yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `m_krk`.

##### PC Server:

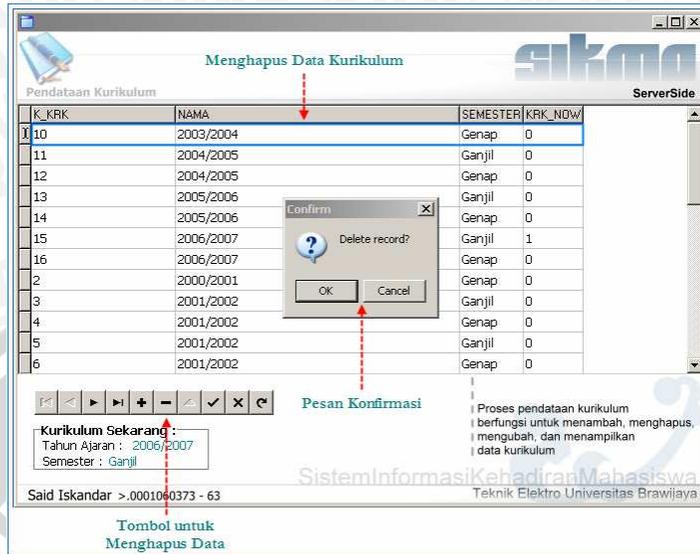
- *Server* basis data kehadiran dijalankan untuk melihat data kurikulum sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_krk ORDER BY k_krk ROWS 5;
```

#### C. Hasil Pengujian dan Analisis

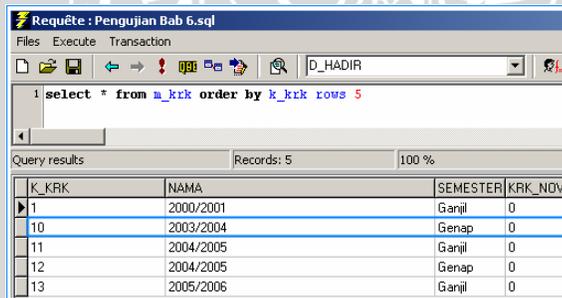
- Administrator akan menghapus *record* data dengan `K_KRK` '10', `NAMA` '2003/2004', `SEMESTER` 'Genap', dan `KRK_NOW` '0'.

Tampilan antarmuka proses penghapusan data kurikulum ditunjukkan dalam Gambar 6.54.



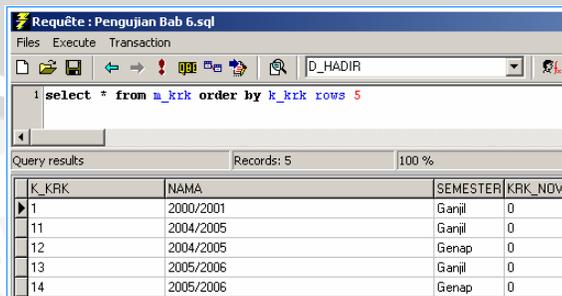
**Gambar 6.54** Antarmuka Proses Penghapusan Data Kurikulum  
Sumber: *Pengujian*

- Data kurikulum sebelum proses penghapusan data kurikulum pada basis data kehadiran ditunjukkan dalam Gambar 6.55.



**Gambar 6.55** Data Kurikulum sebelum Proses Penghapusan Data Kurikulum  
Sumber: *Pengujian*

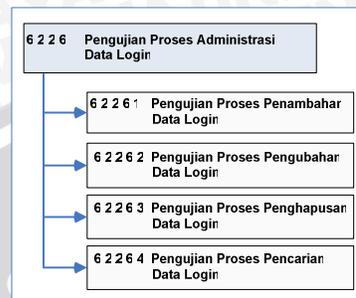
- Data kurikulum setelah proses penghapusan data kurikulum pada basis data kehadiran ditunjukkan dalam Gambar 6.56.



**Gambar 6.56** Data Kurikulum setelah Proses Penghapusan Data Kurikulum  
Sumber: *Pengujian*

### 6.2.2.6 Pengujian Proses Administrasi Data Login

Pengujian proses administrasi data *login* meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data *login* ditunjukkan dalam Gambar 6.57.



Gambar 6.57 Diagram Pohon Pengujian Proses Administrasi Data Login  
Sumber: Perancangan

#### 6.2.2.6.1 Pengujian Proses Penambahan Data Login

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data *login* dapat digunakan untuk menambah data *login* yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Login pada halaman utama aplikasi server.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data *login* baru dan memasukkan data *login* baru ke dalam tabel data *login*.
- Administrator menyimpan data *login* baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (✓).

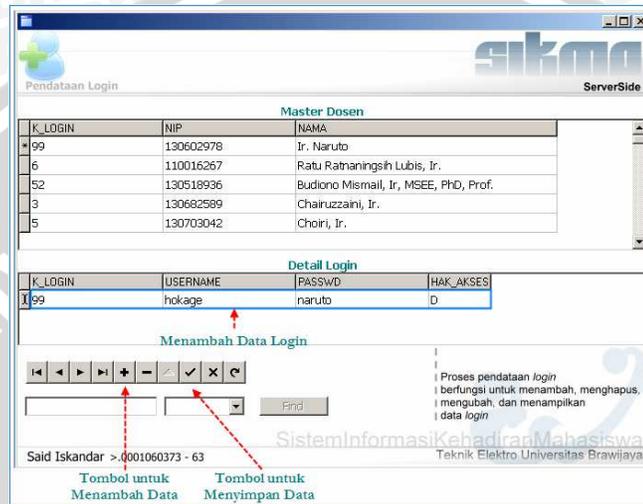
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data *login* baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_login ORDER BY k_login desc ROWS 5;
```

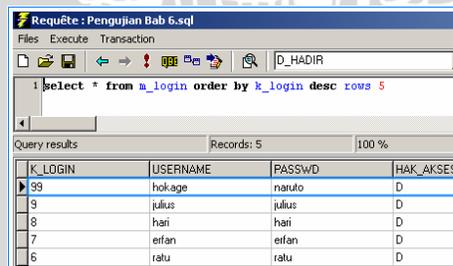
**C. Hasil Pengujian dan Analisis**

- Administrator mengisi tabel data *login* dengan masukan data *login* baru yang terdiri atas kode *login*, *username*, *password*, dan hak akses. Administrator mengisi *record* kosong dengan K\_LOGIN '99', USERNAME 'hokage', PASSWORD 'naruto', dan HAK\_AKSES 'D'. Tampilan antarmuka proses penambahan data *login* ditunjukkan dalam Gambar 6.58.



**Gambar 6.58** Antarmuka Proses Penambahan Data *Login*  
Sumber: *Pengujian*

- Data *login* yang telah ditambahkan akan disimpan pada tabel *m\_login*. Hasil proses penambahan data *login* baru pada basis data kehadiran ditunjukkan dalam Gambar 6.59.



**Gambar 6.59** Hasil Penambahan Data *Login* pada Basis Data Kehadiran  
Sumber: *Pengujian*

**6.2.2.6.2 Pengujian Proses Pengubahan Data *Login***

**A. Tujuan**

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data *login* dapat digunakan untuk mengubah data *login* yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Login pada halaman utama aplikasi server.
- Administrator mengubah data *login* dengan menekan tombol navigasi bertanda (▲) dan memasukkan data *login* yang akan diubah ke dalam tabel data *login*.
- Administrator menyimpan data *login* pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

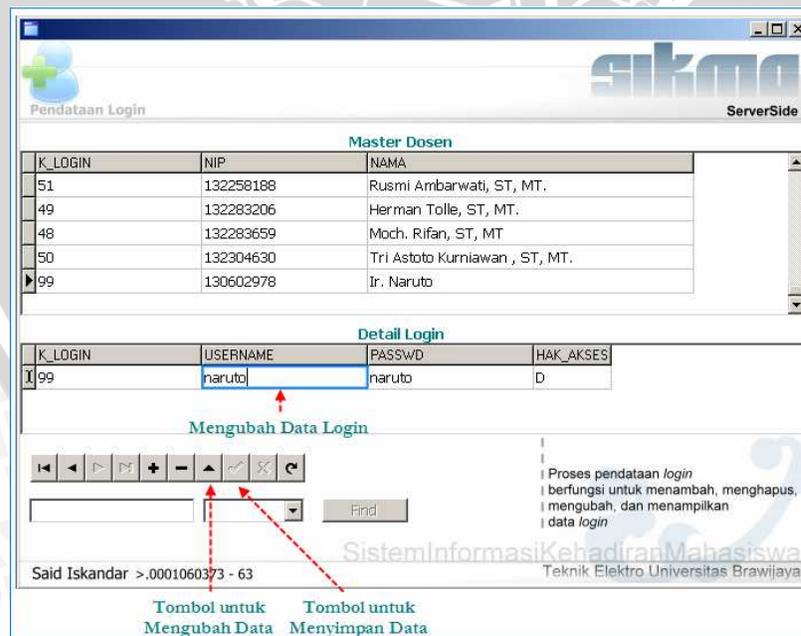
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data *login* yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_login ORDER BY k_login desc ROWS 5;
```

## C. Hasil Pengujian dan Analisis

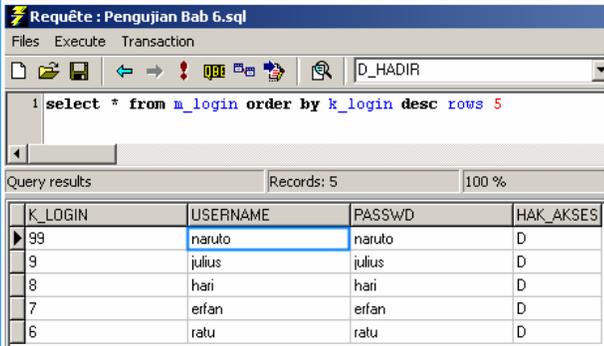
- Administrator mengubah isi tabel data *login* dengan masukan data *login* berupa *username*. Administrator mengganti USERNAME 'hokage' dengan USERNAME 'naruto'. Tampilan antarmuka proses pengubahan data *login* ditunjukkan dalam Gambar 6.60.



Gambar 6.60 Antarmuka Proses Pengubahan Data Login

Sumber: Pengujian

- Data *login* yang telah diubah akan disimpan pada tabel *m\_login*. Hasil proses pengubahan data *login* pada basis data kehadiran ditunjukkan dalam Gambar 6.61.



K_LOGIN	USERNAME	PASSWD	HAK_AKSES
99	naruto	naruto	D
9	julius	julius	D
8	hari	hari	D
7	erfan	erfan	D
6	ratu	ratu	D

**Gambar 6.61** Hasil Pengubahan Data *Login* pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.6.3 Pengujian Proses Penghapusan Data *Login*

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data *login* dapat digunakan untuk menghapus data *login* yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Login pada halaman utama aplikasi server.
- Administrator dapat menghapus data *login* dengan menekan tombol navigasi bertanda (–) setelah memilih data *login* yang akan dihapus.
- Administrator menekan tombol OK pada form pesan konfirmasi untuk menghapus data pada tabel *m\_login*.

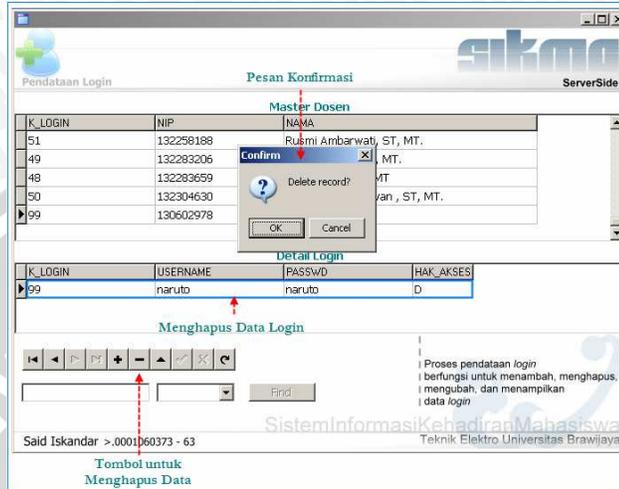
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data *login* sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_login ORDER BY k_login rows 5;
```

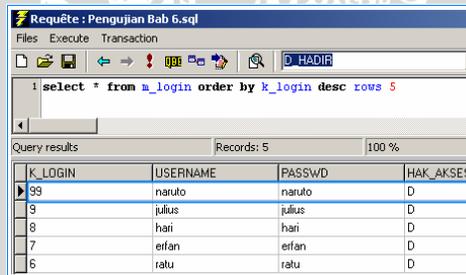
**C. Hasil Pengujian dan Analisis**

- Administrator akan menghapus *record* data dengan K\_LOGIN ‘99’, USERNAME ‘naruto’, PASSWORD ‘naruto’, dan HAK\_AKSES ‘D’. Tampilan antarmuka proses penghapusan data *login* ditunjukkan dalam Gambar 6.62.



**Gambar 6.62** Antarmuka Proses Penghapusan Data *Login*  
Sumber: *Pengujian*

- Data *login* sebelum proses penghapusan data *login* pada basis data kehadiran ditunjukkan dalam Gambar 6.63.



**Gambar 6.63** Data *Login* sebelum Proses Penghapusan Data *Login*  
Sumber: *Pengujian*

- Data *login* setelah proses penghapusan data *login* pada basis data kehadiran ditunjukkan dalam Gambar 6.64.



**Gambar 6.64** Data *Login* setelah Proses Penghapusan Data *Login*  
Sumber: *Pengujian*

#### 6.2.2.6.4 Pengujian Proses Pencarian Data *Login*

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data *login* dapat digunakan untuk mencari data *login* yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.
- Administrator dapat menekan tombol cari atau tombol *enter* untuk mengeksekusi perintah pencarian.

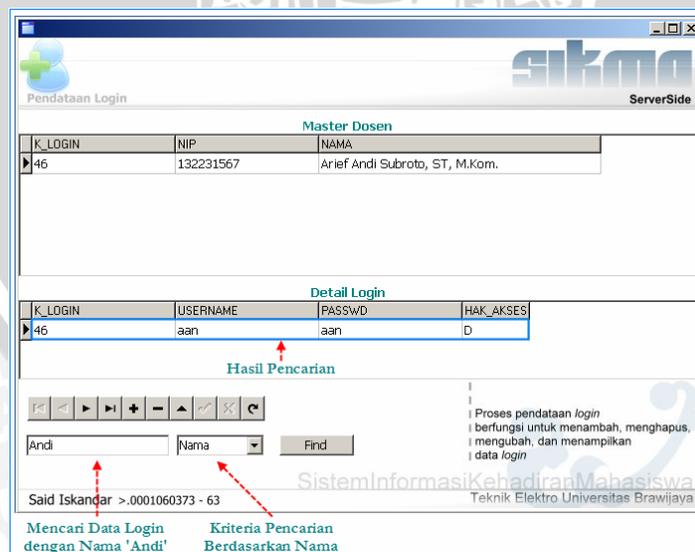
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat hasil pencarian data *login* dengan memberikan *query* sebagai berikut:

```
SELECT * FROM m_dosen WHERE nama CONTAINING 'Andi';
```

##### C. Hasil Pengujian dan Analisis

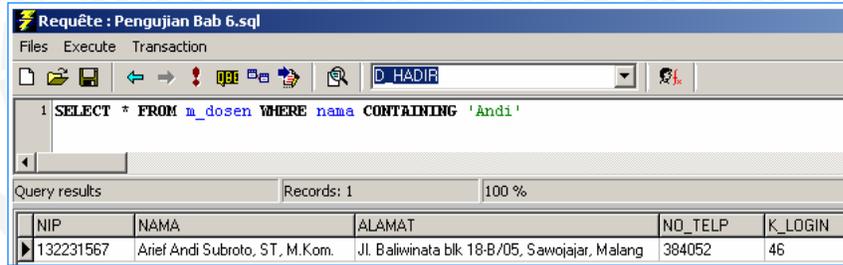
- Administrator memilih kriteria pencarian berupa NIP dan mengisi kata kunci pencarian dengan NIP '978' pada masukan pencarian. Tampilan antarmuka proses pencarian data dosen ditunjukkan dalam Gambar 6.65.



Gambar 6.65 Antarmuka Proses Pencarian Data *Login*

Sumber: *Pengujian*

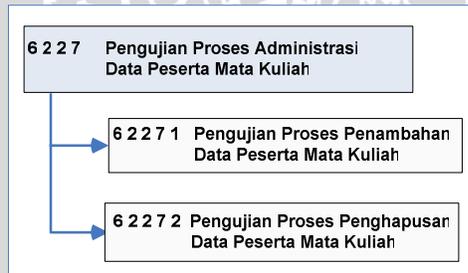
- Hasil proses pencarian menunjukkan bahwa data dosen dengan nama mengandung 'Andi' telah tersimpan pada basis data kehadiran. Hasil proses pencarian data dosen ditunjukkan dalam Gambar 6.66.



**Gambar 6.66** Hasil Pencarian Data *Login* pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.7 Pengujian Proses Administrasi Data Peserta Mata Kuliah

Pengujian proses administrasi data peserta mata kuliah meliputi pengujian proses penambahan data dan penghapusan data. Diagram pohon pengujian proses administrasi data peserta mata kuliah ditunjukkan dalam Gambar 6.67.



**Gambar 6.67** Diagram Pohon Pengujian Proses Administrasi Data Peserta Mata Kuliah  
Sumber: *Perancangan*

#### 6.2.2.7.1 Pengujian Proses Penambahan Data Peserta Mata Kuliah

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data peserta mata kuliah dapat digunakan untuk menambah data peserta mata kuliah yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Peserta Kuliah pada halaman utama aplikasi server.

- Administrator menekan tombol *mouse* sebelah kiri sebanyak dua kali pada tabel jadwal mata kuliah untuk menambah data peserta mata kuliah baru dan memasukkan data peserta mata kuliah baru ke dalam tabel data peserta mata kuliah.
- Administrator menyimpan data peserta mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda Simpan.

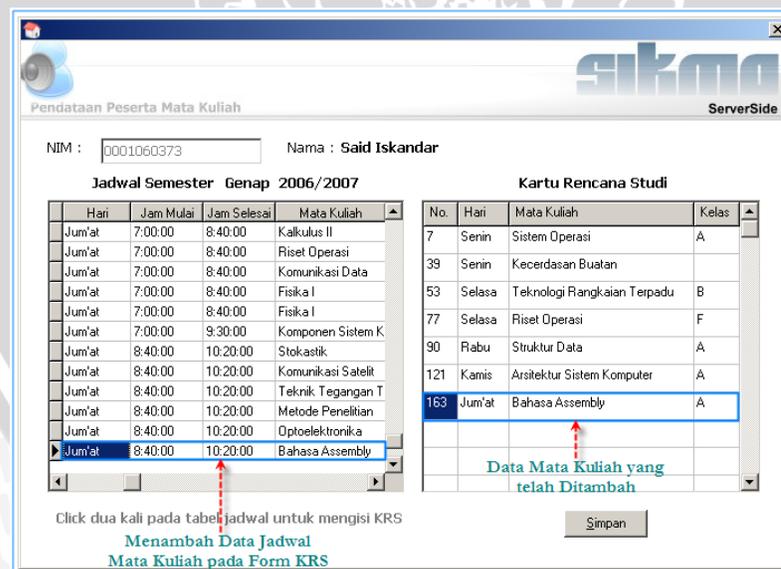
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data peserta mata kuliah baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_peserta_mk;
```

### C. Hasil Pengujian dan Analisis

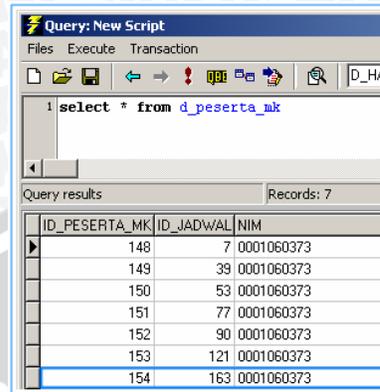
- Administrator mengisi tabel data peserta mata kuliah dengan masukan data peserta mata kuliah baru yang terdiri atas nomor, hari, mata kuliah, dan kelas mata kuliah. Administrator mengisi *record* kosong dengan No. '163', HARI 'Jum'at', Mata Kuliah 'Bahasa Assembly', dan Kelas 'A'. Tampilan antarmuka proses penambahan data peserta mata kuliah ditunjukkan dalam Gambar 6.68.



**Gambar 6.68** Antarmuka Proses Penambahan Data Peserta Mata Kuliah  
Sumber: *Pengujian*

- Data peserta mata kuliah yang telah ditambahkan akan disimpan pada tabel `d_peserta_mk`.

Hasil proses penambahan data peserta mata kuliah baru pada basis data kehadiran ditunjukkan dalam Gambar 6.69.



ID_PESERTA_MK	ID_JADWAL	NIM
148	7	0001060373
149	39	0001060373
150	53	0001060373
151	77	0001060373
152	90	0001060373
153	121	0001060373
154	163	0001060373

**Gambar 6.69** Hasil Penambahan Data Peserta Mata Kuliah pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.7.2 Pengujian Proses Penghapusan Data Peserta Mata Kuliah

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data peserta mata kuliah dapat digunakan untuk menghapus data peserta mata kuliah yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Peserta Kuliah pada halaman utama aplikasi *server*.
- Administrator dapat menghapus data peserta mata kuliah dengan menekan tombol *mouse* sebelah kiri pada tabel Kartu Rencana Studi (KRS) sebanyak dua kali setelah memilih data peserta mata kuliah yang akan dihapus.

PC Server:

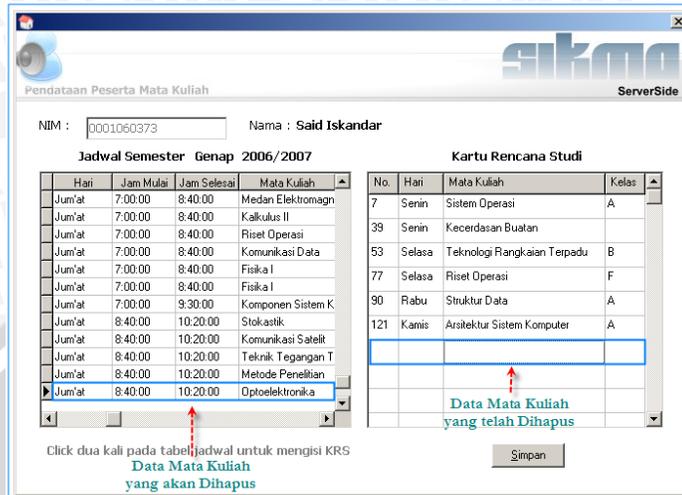
- *Server* basis data kehadiran dijalankan untuk melihat data peserta mata kuliah sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_peserta_mk;
```

#### C. Hasil Pengujian dan Analisis

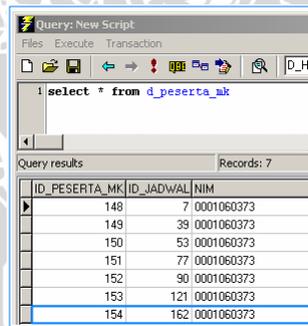
- Administrator akan menghapus *record* data dengan No. '162', HARI 'Jum'at', Mata Kuliah 'Optoelektronika, dan Kelas 'A'.

Tampilan antarmuka proses penghapusan data peserta mata kuliah ditunjukkan dalam Gambar 6.70.



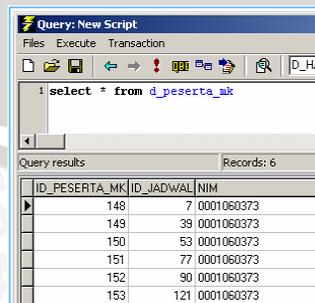
**Gambar 6.70** Antarmuka Proses Penghapusan Data Peserta Mata Kuliah  
Sumber: *Pengujian*

- Data peserta mata kuliah sebelum proses penghapusan data peserta mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.71.



**Gambar 6.71** Data Peserta Mata Kuliah sebelum Proses Penghapusan Data Peserta Mata Kuliah  
Sumber: *Pengujian*

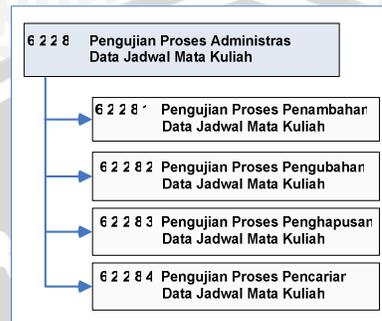
- Data peserta mata kuliah setelah proses penghapusan data peserta mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.72.



**Gambar 6.72** Data Peserta Mata Kuliah setelah Proses Penghapusan Data Peserta Mata Kuliah  
Sumber: *Pengujian*

### 6.2.2.8 Pengujian Proses Administrasi Data Jadwal Mata Kuliah

Pengujian proses administrasi data jadwal mata kuliah meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data jadwal mata kuliah ditunjukkan dalam Gambar 6.73.



**Gambar 6.73** Diagram Pohon Pengujian Proses Administrasi Data Jadwal Mata Kuliah  
Sumber: *Perancangan*

#### 6.2.2.8.1 Pengujian Proses Penambahan Data Jadwal Mata Kuliah

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data jadwal mata kuliah dapat digunakan untuk menambah data jadwal mata kuliah yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Jadwal Kuliah pada halaman utama aplikasi *server*.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data jadwal mata kuliah baru dan memasukkan data jadwal mata kuliah baru ke dalam tabel data jadwal mata kuliah.
- Administrator menyimpan data jadwal mata kuliah baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

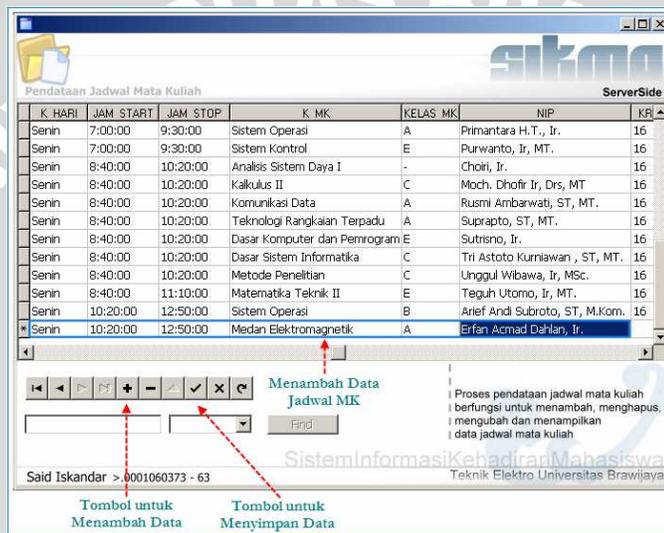
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data jadwal mata kuliah baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_jadwal ORDER BY id_jadwal DESC ROWS 5;
```

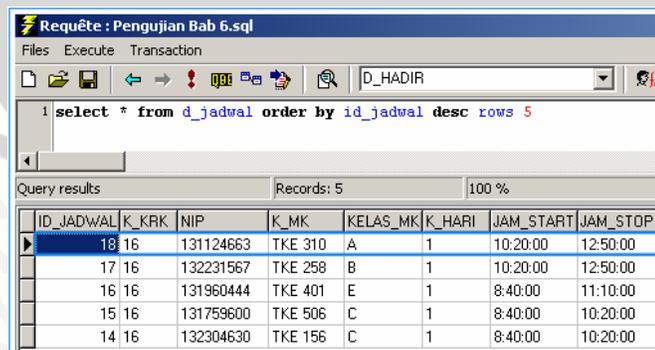
**C. Hasil Pengujian dan Analisis**

- Administrator mengisi tabel data jadwal mata kuliah dengan masukan data jadwal mata kuliah baru yang terdiri atas id jadwal mata kuliah, hari, jam mulai, jam selesai, kode mata kuliah, kelas mata kuliah, dan dosen pengajar. Administrator mengisi *record* kosong dengan ID\_JADWAL '18', K\_HARI 'Senin', JAM\_START '10:20:00', JAM\_STOP '12:50:00', K\_MK 'Medan Elektromagnetik', KELAS\_MK 'A', dan NIP 'Erfan Achmad Dahlan, Ir.'. Tampilan antarmuka proses penambahan data jadwal mata kuliah ditunjukkan dalam Gambar 6.74.



**Gambar 6.74** Antarmuka Proses Penambahan Data Jadwal Mata Kuliah  
Sumber: *Pengujian*

- Data jadwal mata kuliah yang telah ditambahkan akan disimpan pada tabel *d\_jadwal*. Hasil proses penambahan data jadwal mata kuliah baru pada basis data kehadiran ditunjukkan dalam Gambar 6.75.



**Gambar 6.75** Hasil Penambahan Data Jadwal Mata Kuliah pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.8.2 Pengujian Proses Pengubahan Data Jadwal Mata Kuliah

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data jadwal mata kuliah dapat digunakan untuk mengubah data jadwal mata kuliah yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Jadwal Kuliah pada halaman utama aplikasi server.
- Administrator mengubah data jadwal mata kuliah dengan menekan tombol navigasi bertanda (▲) dan memasukkan data jadwal mata kuliah yang akan diubah ke dalam tabel data jadwal mata kuliah.
- Administrator menyimpan data jadwal mata kuliah pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

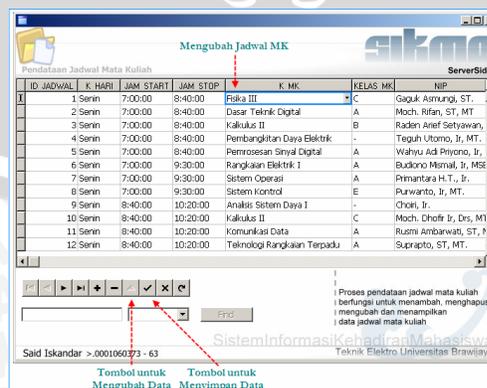
PC Server:

- Server basis data kehadiran dijalankan untuk melihat data jadwal mata kuliah yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_jadwal ORDER BY id_jadwal ROWS 5;
```

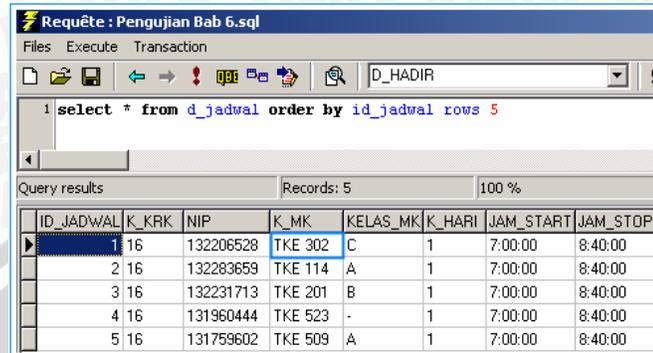
#### C. Hasil Pengujian dan Analisis

- Administrator mengubah isi tabel data jadwal mata kuliah dengan masukan data jadwal mata kuliah berupa kode jadwal mata kuliah. Administrator mengganti K\_MK 'Bahan Elektrik dengan K\_MK 'Fisika III' pada *record* pertama. Tampilan antarmuka proses pengubahan data jadwal mata kuliah ditunjukkan dalam Gambar 6.76.



Gambar 6.76 Antarmuka Proses Pengubahan Data Jadwal Mata Kuliah  
Sumber: Pengujian

- Data jadwal mata kuliah yang telah diubah akan disimpan pada tabel `d_jadwal`. Hasil proses pengubahan data jadwal mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.77.



ID_JADWAL	K_KRK	NIP	K_MK	KELAS_MK	K_HARI	JAM_START	JAM_STOP
1	16	132206528	TKE 302	C	1	7:00:00	8:40:00
2	16	132283659	TKE 114	A	1	7:00:00	8:40:00
3	16	132231713	TKE 201	B	1	7:00:00	8:40:00
4	16	131960444	TKE 523	.	1	7:00:00	8:40:00
5	16	131759602	TKE 509	A	1	7:00:00	8:40:00

Gambar 6.77 Hasil Pengubahan Data Jadwal Mata Kuliah pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.8.3 Pengujian Proses Penghapusan Data Jadwal Mata Kuliah

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data jadwal mata kuliah dapat digunakan untuk menghapus data jadwal mata kuliah yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Jadwal Kuliah pada halaman utama aplikasi *server*.
- Administrator dapat menghapus data jadwal mata kuliah dengan menekan tombol navigasi bertanda (–) setelah memilih data jadwal mata kuliah yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel `d_jadwal`.

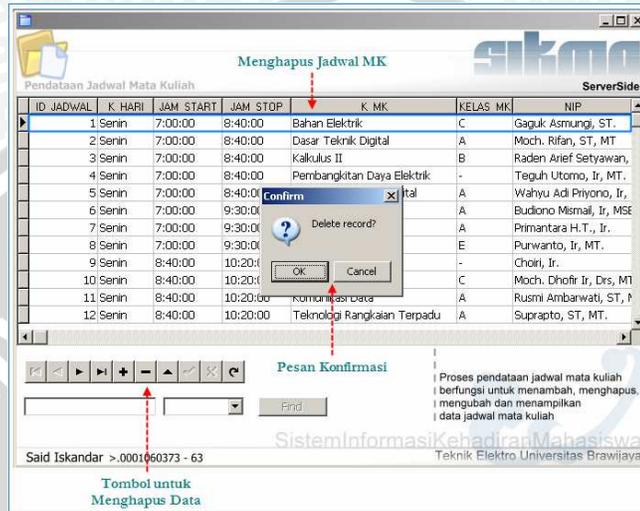
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data jadwal mata kuliah sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_jadwal ORDER BY id_jadwal ROWS 5;
```

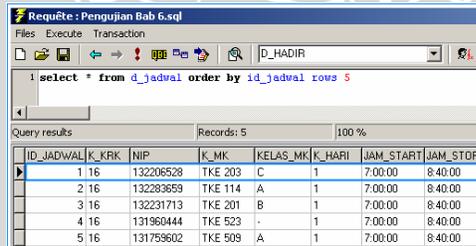
**C. Hasil Pengujian dan Analisis**

- Administrator akan menghapus *record* data dengan ID\_JADWAL '1', K\_HARI 'Senin', K\_HARI 'Senin', K\_MK 'Bahan Elektrik', dan KELAS\_MK 'C'. Tampilan antarmuka proses penghapusan data jadwal mata kuliah ditunjukkan dalam Gambar 6.78.



**Gambar 6.78** Antarmuka Proses Penghapusan Data Jadwal Mata Kuliah  
Sumber: *Pengujian*

- Data jadwal mata kuliah sebelum proses penghapusan data jadwal mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.79.



**Gambar 6.79** Data Jadwal Mata Kuliah sebelum Proses Penghapusan Data Jadwal Mata Kuliah  
Sumber: *Pengujian*

- Data jadwal mata kuliah setelah proses penghapusan data jadwal mata kuliah pada basis data kehadiran ditunjukkan dalam Gambar 6.80.



**Gambar 6.80** Data Jadwal Mata Kuliah setelah Proses Penghapusan Data Jadwal Mata Kuliah  
Sumber: *Pengujian*

#### 6.2.2.8.4 Pengujian Proses Pencarian Data Jadwal Mata Kuliah

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data jadwal mata kuliah dapat digunakan untuk mencari data jadwal mata kuliah yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.
- Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian.

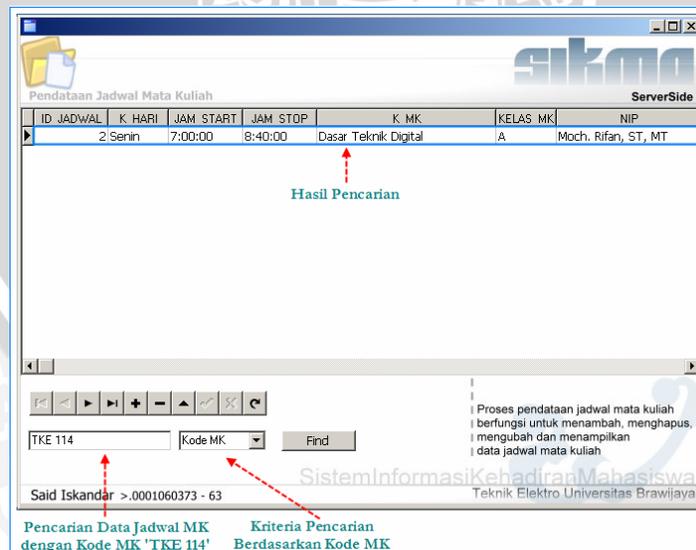
PC Server:

- Server basis data kehadiran dijalankan untuk melihat hasil pencarian data jadwal mata kuliah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_jadwal WHERE nama CONTAINING 'TKE 114';
```

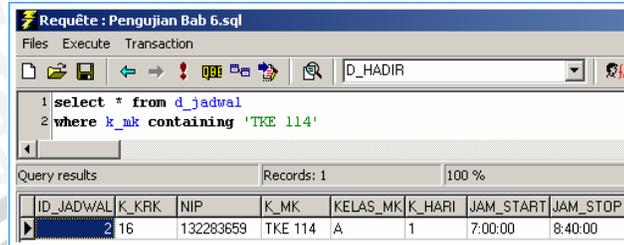
##### C. Hasil Pengujian dan Analisis

- Administrator memilih kriteria pencarian berupa K\_MK dan mengisi kata kunci pencarian dengan K\_MK 'TKE 114'. Tampilan antarmuka proses pencarian data jadwal mata kuliah ditunjukkan dalam Gambar 6.81.



**Gambar 6.81** Antarmuka Proses Pencarian Data Jadwal Mata Kuliah  
Sumber: *Pengujian*

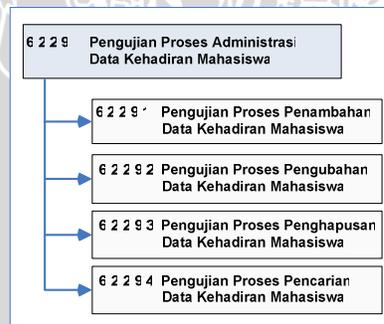
- Hasil proses pencarian menunjukkan bahwa data jadwal mata kuliah dengan K\_MK 'TKE 114' telah tersimpan pada basis data kehadiran. Hasil proses pencarian data jadwal mata kuliah ditunjukkan dalam Gambar 6.82.



Gambar 6.82 Hasil Proses Pencarian Data Jadwal Mata Kuliah pada Basis Data Kehadiran  
 Sumber: Pengujian

### 6.2.2.9 Pengujian Proses Administrasi Data Kehadiran Mahasiswa

Pengujian proses administrasi data kehadiran mahasiswa meliputi pengujian proses penambahan data, pengubahan data, penghapusan data, dan pencarian data. Diagram pohon pengujian proses administrasi data kehadiran mahasiswa ditunjukkan dalam Gambar 6.83.



Gambar 6.83 Diagram Pohon Pengujian Proses Administrasi Data Kehadiran Mahasiswa  
 Sumber: Perancangan

#### 6.2.2.9.1 Pengujian Proses Penambahan Data Kehadiran Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kehadiran mahasiswa dapat digunakan untuk menambah data kehadiran mahasiswa yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Kehadiran pada halaman utama aplikasi *server*.
- Administrator menekan tombol navigasi bertanda (+) untuk menambah data kehadiran mahasiswa baru dan memasukkan data kehadiran mahasiswa baru ke dalam tabel data kehadiran mahasiswa.
- Administrator menyimpan data kehadiran mahasiswa baru pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

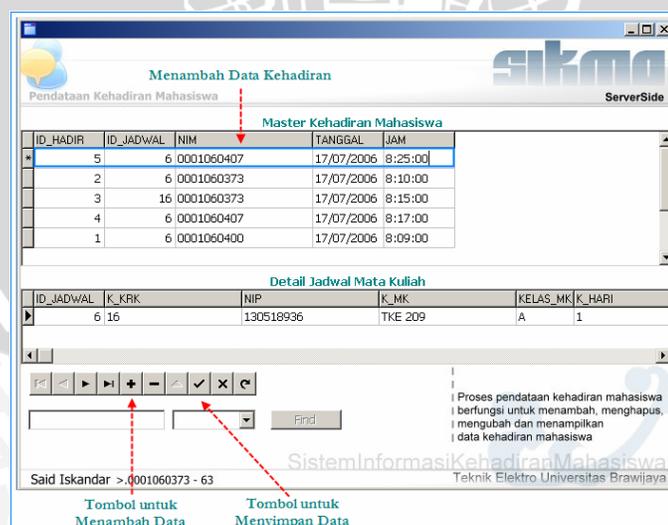
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data kehadiran mahasiswa baru dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_hadir ORDER BY id_hadir;
```

## C. Hasil Pengujian dan Analisis

- Administrator mengisi tabel data kehadiran mahasiswa dengan masukan data kehadiran mahasiswa baru yang terdiri atas id hadir, id jadwal, nim, tanggal, dan jam. Administrator mengisi *record* kosong dengan ID\_HADIR '5', ID\_JADWAL '6', dan NIM '0001060407', TANGGAL '17/07/2006', dan JAM '8:25:00'. Tampilan antarmuka proses penambahan data kehadiran mahasiswa ditunjukkan dalam Gambar 6.84.



Gambar 6.84 Antarmuka Proses Penambahan Data Kehadiran Mahasiswa  
Sumber: Pengujian

- Data kehadiran mahasiswa yang telah ditambahkan akan disimpan pada tabel `d_hadir`. Hasil proses penambahan data kehadiran mahasiswa baru pada basis data kehadiran ditunjukkan dalam Gambar 6.85.



ID_HADIR	ID_JADWAL	NIM	TANGGAL	JAM
1	6	0001060400	17/07/2006	8:09:00
2	6	0001060373	17/07/2006	8:10:00
3	16	0001060373	17/07/2006	8:15:00
4	6	0001060407	17/07/2006	8:17:00
5	6	0001060407	17/07/2006	8:25:00

**Gambar 6.85** Hasil Penambahan Data Kehadiran Mahasiswa pada Basis Data Kehadiran  
Sumber: *Pengujian*

### 6.2.2.9.2 Pengujian Proses Perubahan Data Kehadiran Mahasiswa

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kehadiran mahasiswa dapat digunakan untuk mengubah data kehadiran mahasiswa yang terdapat pada basis data kehadiran.

#### B. Prosedur Pengujian

*PC Client:*

- Administrator memilih menu Pendataan Kehadiran pada halaman utama aplikasi *server*.
- Administrator mengubah data kehadiran mahasiswa dengan menekan tombol navigasi bertanda (▲) dan memasukkan data kehadiran mahasiswa yang akan diubah ke dalam tabel data kehadiran mahasiswa.
- Administrator menyimpan data kehadiran mahasiswa pada basis data kehadiran dengan menekan tombol navigasi bertanda (√).

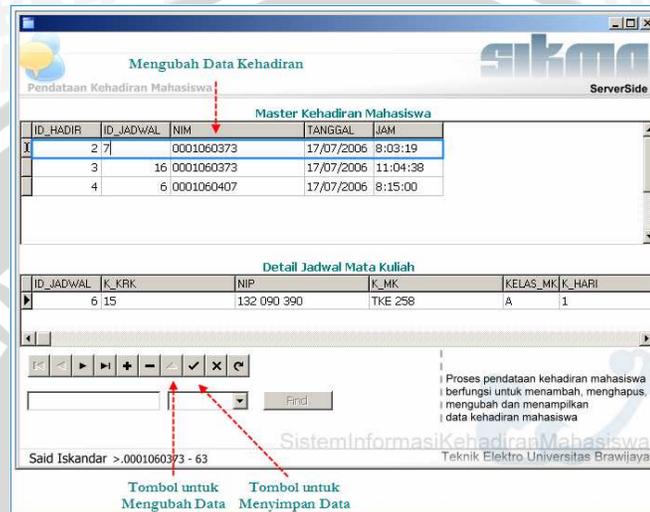
*PC Server:*

- *Server* basis data kehadiran dijalankan untuk melihat data kehadiran mahasiswa yang telah diubah dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_hadir ORDER BY id_hadir;
```

### C. Hasil Pengujian dan Analisis

- Administrator mengubah isi tabel data kehadiran mahasiswa dengan masukan data kehadiran mahasiswa berupa id jadwal. Administrator mengganti ID\_JADWAL '1' dengan ID\_JADWAL '7' pada *record* pertama. Tampilan antarmuka proses perubahan data kehadiran mahasiswa ditunjukkan dalam Gambar 6.86.



Gambar 6.86 Antarmuka Proses Perubahan Data Kehadiran Mahasiswa  
Sumber: Pengujian

- Data kehadiran mahasiswa yang telah diubah akan disimpan pada tabel *d\_hadir*. Hasil proses perubahan data kehadiran mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.87.



Gambar 6.87 Hasil Perubahan Data Kehadiran Mahasiswa pada Basis Data Kehadiran  
Sumber: Pengujian

#### 6.2.2.9.3 Pengujian Proses Penghapusan Data Kehadiran Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kehadiran mahasiswa dapat digunakan untuk menghapus data kehadiran mahasiswa yang terdapat pada basis data kehadiran.

## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pendataan Kehadiran pada halaman utama aplikasi *server*.
- Administrator dapat menghapus data kehadiran mahasiswa dengan menekan tombol navigasi bertanda (–) setelah memilih data kehadiran mahasiswa yang akan dihapus.
- Administrator menekan tombol OK pada *form* pesan konfirmasi untuk menghapus data pada tabel *d\_hadir*.

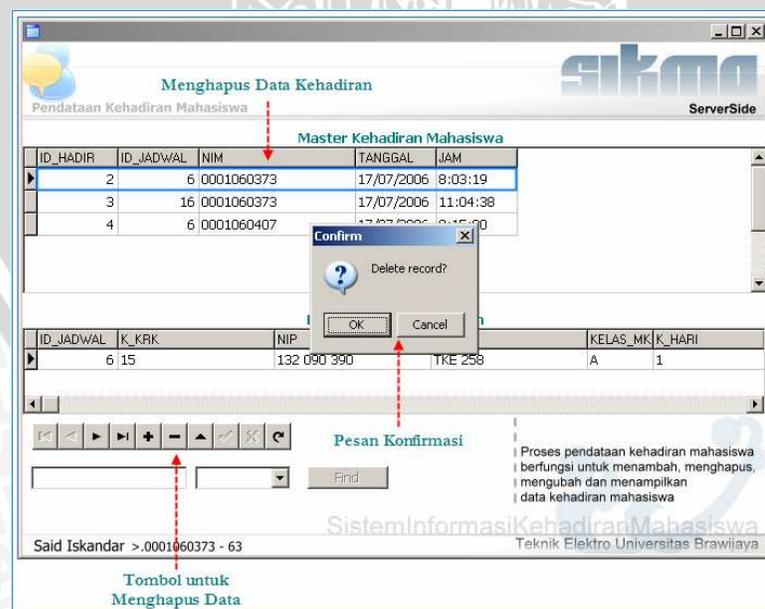
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat data kehadiran mahasiswa sebelum dan setelah proses penghapusan data dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_hadir ORDER BY id_hadir;
```

## C. Hasil Pengujian dan Analisis

- Administrator akan menghapus *record* data dengan ID\_JADWAL '2', ID\_JADWAL '6', NIM '0001060373', TANGGAL '17/17/2006', dan JAM '8:03:19'. Tampilan antarmuka proses penghapusan data kehadiran mahasiswa ditunjukkan dalam Gambar 6.88.



**Gambar 6.88** Antarmuka Proses Penghapusan Data Kehadiran Mahasiswa  
Sumber: *Pengujian*

- Data kehadiran mahasiswa sebelum proses penghapusan data kehadiran mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.89.

The screenshot shows a SQL query window titled 'Requête : Pengujian Bab 6.sql'. The query is 'select \* from d\_hadir order by id\_hadir'. The results table has 3 records:

ID_HADIR	ID_JADWAL	NIM	TANGGAL	JAM
2	6	0001060373	17/07/2006	8:03:19
3	16	0001060373	17/07/2006	11:04:38
4	6	0001060407	17/07/2006	8:15:00

**Gambar 6.89** Data Kehadiran Mahasiswa sebelum Proses Penghapusan Data Kehadiran Mahasiswa  
Sumber: *Pengujian*

- Data kehadiran mahasiswa setelah proses penghapusan data kehadiran mahasiswa pada basis data kehadiran ditunjukkan dalam Gambar 6.90.

The screenshot shows a SQL query window titled 'Requête : Pengujian Bab 6.sql'. The query is 'select \* from d\_hadir order by id\_hadir'. The results table has 2 records:

ID_HADIR	ID_JADWAL	NIM	TANGGAL	JAM
3	16	0001060373	17/07/2006	11:04:38
4	6	0001060407	17/07/2006	8:15:00

**Gambar 6.90** Data Kehadiran Mahasiswa setelah Proses Penghapusan Data Kehadiran Mahasiswa  
Sumber: *Pengujian*

#### 6.2.2.9.4 Pengujian Proses Pencarian Data Kehadiran Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah proses administrasi data kehadiran mahasiswa dapat digunakan untuk mencari data kehadiran mahasiswa yang terdapat pada basis data kehadiran.

##### B. Prosedur Pengujian

PC Client:

- Administrator memilih kriteria pencarian melalui pilihan pencarian dan mengisi kata kunci pencarian pada masukan pencarian.
- Administrator dapat menekan tombol cari atau tombol enter untuk mengeksekusi perintah pencarian.

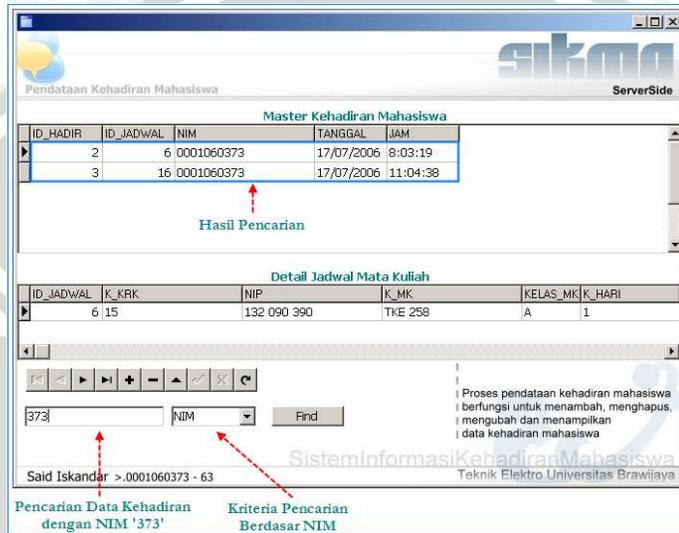
PC Server:

- *Server* basis data kehadiran dijalankan untuk melihat hasil pencarian data kehadiran mahasiswa dengan memberikan *query* sebagai berikut:

```
SELECT * FROM d_hadir WHERE nim CONTAINING '373' ORDER BY id_hadir;
```

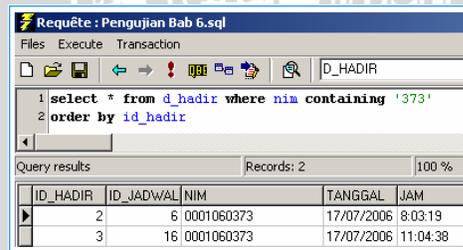
**C. Hasil Pengujian dan Analisis**

- Administrator memilih kriteria pencarian berupa NIM dan mengisi kata kunci pencarian dengan NIM '373' pada masukan pencarian. Tampilan antarmuka proses pencarian data mahasiswa ditunjukkan dalam Gambar 6.91.



**Gambar 6.91** Antarmuka Proses Pencarian Data Kehadiran Mahasiswa  
Sumber: *Pengujian*

- Hasil proses pencarian menunjukkan bahwa data kehadiran mahasiswa dengan NIM '373' telah tersimpan pada basis data kehadiran. Hasil proses pencarian data kehadiran mahasiswa ditunjukkan dalam Gambar 6.92.



**Gambar 6.92** Hasil Proses Pencarian Data Kehadiran Mahasiswa pada Basis Data Kehadiran  
Sumber: *Pengujian*

**6.2.2.10 Pengujian Proses Pencetakan Kartu Kehadiran Mahasiswa**

**A. Tujuan**

- Pengujian dilakukan untuk mengetahui apakah aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak Kartu Kehadiran Mahasiswa (KKM).

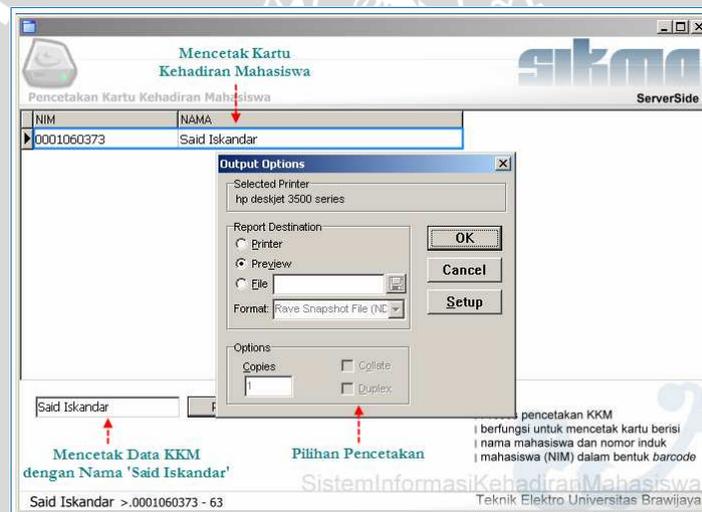
## B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pencetakan KKM pada halaman utama aplikasi server.
- Administrator mengisi kata kunci pencarian dan form pilihan pencetakan akan ditampilkan setelah administrator menekan tombol Print.
- Administrator dapat memilih untuk mencetak ke printer, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk file.

## C. Hasil Pengujian dan Analisis

- Tampilan antarmuka aplikasi proses pencetakan KKM ditunjukkan dalam Gambar 6.93.



Gambar 6.93 Antarmuka Proses Pencetakan KKM

Sumber: Pengujian

- Tampilan hasil Kartu Kehadiran Mahasiswa (KKM) berdasarkan masukan data nim ditunjukkan dalam Gambar 6.94.



Gambar 6.94 Kartu Kehadiran Mahasiswa

Sumber: Pengujian

- Aplikasi proses cetak kartu pada aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak Kartu Kehadiran Mahasiswa.

### 6.2.2.11 Pengujian Proses Pencetakan Kartu Kehadiran Dosen

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak Kartu Kehadiran Dosen (KKD).

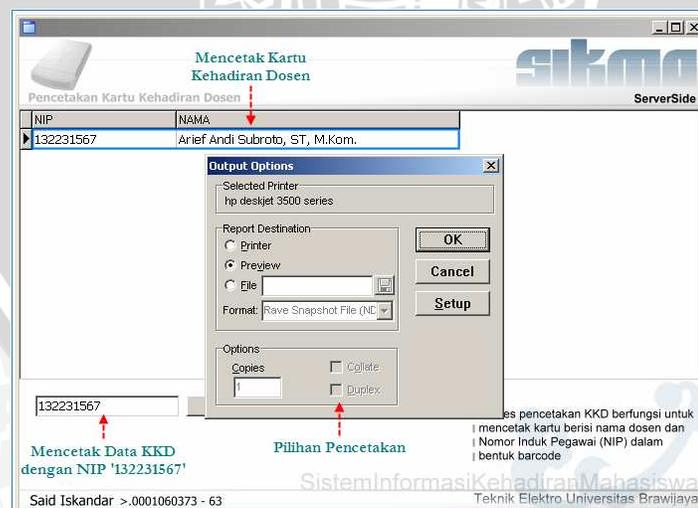
#### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pencetakan KKD pada halaman utama aplikasi server.
- Administrator mengisi kata kunci pencarian dan *form* pilihan pencetakan akan ditampilkan setelah administrator menekan tombol Print.
- Administrator dapat memilih untuk mencetak ke *printer*, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk *file*.

#### C. Hasil Pengujian dan Analisis

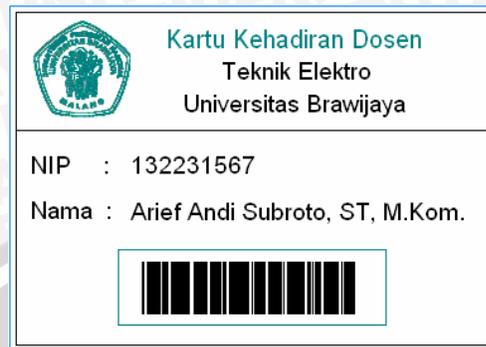
- Tampilan antarmuka aplikasi proses pencetakan KKD ditunjukkan dalam Gambar 6.95.



**Gambar 6.95** Antarmuka Proses Pencetakan KKD

Sumber: *Pengujian*

- Tampilan hasil Kartu Kehadiran Dosen (KKD) berdasarkan masukan data nip ditunjukkan dalam Gambar 6.96.



**Gambar 6.96** Kartu Kehadiran Dosen  
Sumber: *Pengujian*

- Aplikasi proses cetak kartu pada aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak Kartu Kehadiran Dosen.

#### 6.2.2.12 Pengujian Proses Pencetakan *Report* Kehadiran Mahasiswa

##### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak *report* kehadiran mahasiswa.

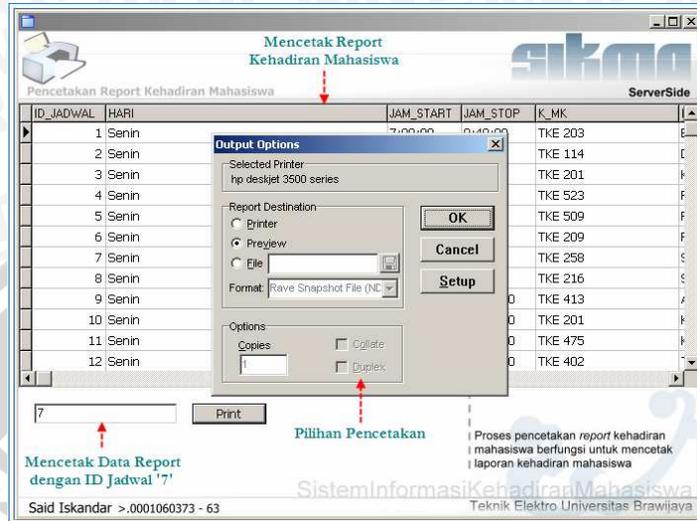
##### B. Prosedur Pengujian

PC Client:

- Administrator memilih menu Pencetakan Report pada halaman utama aplikasi *server*.
- Administrator mengisi kata kunci pencarian dan *form* pilihan pencetakan akan ditampilkan setelah administrator menekan tombol Print.
- Administrator dapat memilih untuk mencetak ke *printer*, melihat hasil pencetakan di layar komputer, atau menyimpan hasil pencetakan dalam bentuk *file*.

**C. Hasil Pengujian dan Analisis**

- Tampilan antarmuka aplikasi proses pencetakan *report* kehadiran mahasiswa ditunjukkan dalam Gambar 6.97.



**Gambar 6.97** Antarmuka Proses Pencetakan *report* kehadiran mahasiswa  
Sumber: *Pengujian*

- Tampilan hasil *report* kehadiran mahasiswa berdasarkan masukan data id jadwal ditunjukkan dalam Gambar 6.98.

 Universitas Brawijaya Fakultas Teknik Jurusan Teknik Elektro			
<b>Report Kehadiran Mahasiswa</b> Semester : Genap Tahun : 2006/2007			
Mata Kuliah : Sistem Operasi		Dosen Pengajar : Primantara H.T	
Kelas MK : A		Kehadiran Dosen : 4	
NIM	Nama Mahasiswa	Jumlah Hadir	%
0001060272	Ahmad Wildan Nuruzzaman	2	50
0001060301	Dwi Adi Basuki	1	25
0001060373	Said Iskandar	3	75
0001060400	Wawan Rachmanto	4	100
0001060407	Yoga Al Kautzar Putra	3	75

**Gambar 6.98** *Report* Kehadiran Mahasiswa  
Sumber: *Pengujian*

- Aplikasi proses cetak kartu pada aplikasi sistem informasi kehadiran mahasiswa dapat digunakan untuk mencetak *report* kehadiran mahasiswa.

**6.2.3 Pengujian Aplikasi Client**

Pengujian aplikasi *client* meliputi pengujian proses *login client* yang dilakukan oleh dosen dan pengujian pencatatan kehadiran mahasiswa yang berfungsi untuk menyimpan data kehadiran mahasiswa.

### 6.2.3.1 Pengujian Proses *Login Client*

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah dosen dapat melakukan proses *login* pada aplikasi *client*.

#### B. Prosedur Pengujian

PC *Client*:

- *Window command prompt* dijalankan dengan memberikan perintah sebagai berikut:

Start | Run... | Open: cmd.exe

- Aplikasi *client* dijalankan dengan memanggil:

G:\! Delphi\delphi\_6\client.exe

- Melakukan proses *login user* aplikasi *client* dengan *nip* 132090390.
- Melakukan proses *logout* pada aplikasi *client*.

#### C. Hasil Pengujian dan Analisis

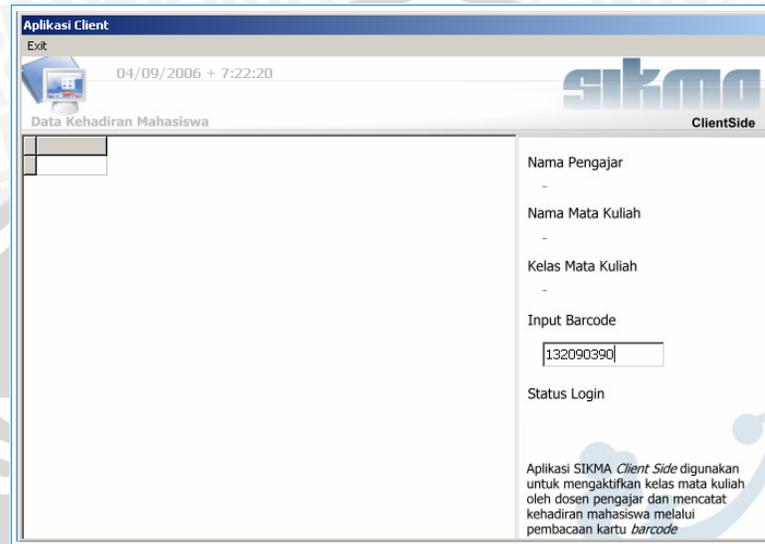
- Halaman antarmuka aplikasi *client* untuk proses *login* ditunjukkan dalam Gambar 6.99.



Gambar 6.99 Halaman antarmuka aplikasi *client* untuk proses *login*

Sumber: Pengujian

- Dosen dengan nip 132090390 melakukan *login* pada tanggal 04/09/2006 dan jam 7:22:20. Proses *login client* menggunakan masukan *nip* 132090390. Tampilan antarmuka proses *login client* yang dilakukan oleh dosen ditunjukkan dalam Gambar 6.100.



**Gambar 6.100** Proses *Login* Dosen  
Sumber: *Pengujian*

- Pesan peringatan akan ditampilkan bila dosen melakukan kesalahan dalam memasukkan *login* atau tidak terdaftar dalam tabel *m\_dosen* pada basis data kehadiran. Pesan kesalahan dalam proses *login* dosen ditunjukkan dalam Gambar 6.101.



**Gambar 6.101** Pesan Peringatan Proses *Login*  
Sumber: *Pengujian*

- Pesan peringatan akan ditampilkan bila proses *login* dosen berhasil tetapi dosen yang bersangkutan tidak mempunyai jadwal mengajar pada hari dan jam saat aplikasi *client* dijalankan. Pesan kesalahan dalam pengaktifan jadwal mengajar ditunjukkan dalam Gambar 6.102.



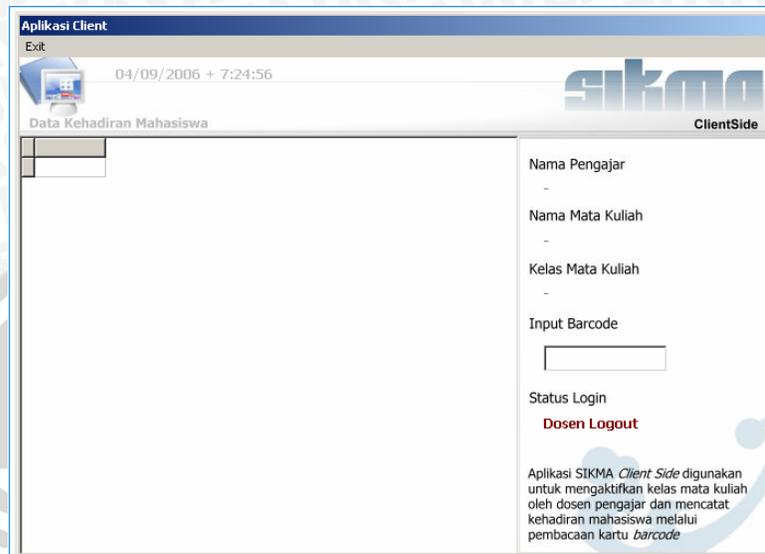
**Gambar 6.102** Pesan Peringatan Jadwal Mengajar  
Sumber: *Pengujian*

- Halaman utama aplikasi *client* akan ditampilkan setelah proses *login* selesai dilakukan. Proses *login* yang berhasil dilakukan oleh dosen ditunjukkan dalam Gambar 6.103.



**Gambar 6.103** Proses *Login* Dosen Berhasil Dilakukan  
Sumber: *Pengujian*

- Proses *logout* pada aplikasi *client* dilakukan apabila dosen ingin menutup aplikasi *client*. Proses *logout* yang berhasil dilakukan oleh dosen ditunjukkan dalam Gambar 6.104.



**Gambar 6.104** Proses *Logout* Dosen Berhasil Dilakukan  
Sumber: *Pengujian*

### 6.2.3.2 Pengujian Pencatatan Kehadiran Mahasiswa

#### A. Tujuan

- Pengujian dilakukan untuk mengetahui apakah mahasiswa dapat melakukan proses pencatatan kehadiran pada aplikasi *client*.

#### B. Prosedur Pengujian

PC *Client*:

- *Window command prompt* dijalankan dengan memberikan perintah sebagai berikut:  
Start | Run... | Open: cmd.exe
- Aplikasi *client* dijalankan dengan memanggil:  
G:\! Delphi\delphi\_6\client.exe
- Melakukan proses pencatatan kehadiran mahasiswa pada aplikasi *client* dengan nim 0001060373.

### C. Hasil Pengujian dan Analisis

- Mahasiswa melakukan pencatatan kehadiran dengan masukan nim. Halaman antarmuka aplikasi *client* untuk proses pencatatan kehadiran ditunjukkan dalam Gambar 6.105.



**Gambar 6.105** Halaman Antarmuka Aplikasi *Client* untuk Proses Pencatatan Kehadiran  
Sumber: *Pengujian*

- Pesan peringatan akan ditampilkan bila mahasiswa tidak terdaftar pada mata kuliah yang aktif pada saat aplikasi *client* dijalankan sesuai dengan tabel `d_peserta_mk`. Pesan kesalahan dalam pengaktifan mata kuliah ditunjukkan dalam Gambar 6.106.



**Gambar 6.106** Pesan Peringatan Mata Kuliah  
Sumber: *Pengujian*

- Proses pencatatan kehadiran mahasiswa menggunakan masukan nim 0001060373. Tampilan antarmuka hasil pencatatan kehadiran mahasiswa yang ditunjukkan dalam Gambar 6.107.



**Gambar 6.107** Halaman Antarmuka Hasil Pencatatan Kehadiran Mahasiswa  
Sumber: *Pengujian*

- Pesan peringatan akan ditampilkan bila mahasiswa melakukan proses pencatatan kehadiran lebih dari satu kali. Pesan kesalahan dalam proses pencatatan kehadiran mahasiswa ditunjukkan dalam Gambar 6.108.



**Gambar 6.108** Pesan Peringatan Pencatatan Kehadiran  
Sumber: *Pengujian*

## BAB VII PENUTUP

### 7.1 Kesimpulan

Sistem Informasi Kehadiran Mahasiswa dapat berfungsi sesuai dengan perancangan dan pengujian yang ditunjukkan melalui :

1. Sistem Informasi Kehadiran Mahasiswa terdiri atas aplikasi *server* dan aplikasi *client*.
2. *Data Flow Diagram* (DFD) Sistem Informasi Kehadiran Mahasiswa telah divalidasi kebenarannya. Hal ini dibuktikan dengan pesan "*Diagram Correct*". Pengujian validitas *Data Flow Diagram* (DFD) menggunakan *software* Visible Analyst 7.5.
3. Proses *Generate Entity Relationship Database* yang berhasil dilakukan dengan menggunakan *software* Sybase PowerDesigner 10 tanpa kesalahan.
4. Proses pengujian waktu akses *query* dilakukan dengan memasukkan data linier pada tabel *d\_hadir*, *d\_jadwal*, *d\_peserta\_mk*, dan *m\_dosen* sebanyak 2000, 4000, 6000, 8000, dan 10000 data *entry*. Perbandingan rata-rata waktu akses *query* per-*record* pada basis data kehadiran ditunjukkan dalam Tabel 7.1.

Tabel 7.1 Tabel rata-rata pengujian waktu akses *query* per-*record*

Tes	Jumlah data	Waktu akses	Waktu akses <i>query</i> per- <i>record</i>
1	2000	0,28 s	0,0001400 s
2	4000	0,55 s	0,0001375 s
3	6000	0,79 s	0,0001316 s
4	8000	1,04 s	0,0001300 s
5	10000	1,31 s	0,0001310 s
Rata-rata Waktu Akses <i>Query</i> Per- <i>Record</i>			0,0001340 s

Sumber: *Pengujian*

Sistem basis data Borland Interbase dapat berjalan stabil ditunjukkan dengan linieritas waktu akses *query* per-*record* terhadap masukan jumlah data yang linier.

5. Hasil pengujian yang dilakukan terhadap koneksi basis data dan waktu akses *query* menunjukkan bahwa Sistem Informasi Kehadiran Mahasiswa dapat berfungsi dengan baik.

## 7.2 Saran

Saran yang dapat diberikan untuk pengembangan Sistem Informasi Kehadiran Mahasiswa dengan menggunakan *barcode* antara lain:

1. Sistem Informasi Kehadiran Mahasiswa dapat dikembangkan melalui integrasi dengan Sistem Informasi Akademis untuk masukan data berupa Kartu Rencana Studi (KRS) pada aplikasi *server* bagian peserta mata kuliah.
2. Keamanan Sistem Informasi Kehadiran Mahasiswa dapat dikembangkan dan tidak hanya menggunakan keamanan standar dari basis data Borland Interbase.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- [CAH-02] Cahyana, Ade. 2002. "Implementasi Teknologi Biometric Untuk Sistem Absensi Perkantoran". Bandung: Pusat Penelitian Informatika (PPI-LIPI).
- [DHA-02] Dharma, Budi Sutedjo. 2002. "Perancangan dan Pembangunan Sistem Informasi". Yogyakarta: Andi.
- [DWI-03] Dwiyoga, Albertus. 2003. "Aplikasi Database Server Interbase". Jakarta: Elex Media Komputindo.
- [FAT-02] Fathansyah, Ir. 2002. "Basis Data". Bandung: Informatika.
- [IDA-05] IDAutomation, Inc. 2005. "IDAutomation Barcode FAQ and Tutorial". Akses dari : <http://www.idautomation.com/barcoding4beginners.html>  
Tanggal akses: 13 Januari 2006.
- [IND-03] Indrawaty, Youllia. 2003. "Pengamanan Pada Penyandian Barcode". Jakarta. ar\_barcode\_1.pdf.
- [INT-01] InterBase 6 Documentation. 2001. "Developer's Guide". Borland Software Corporation. A4-Manual.pdf.
- [IRM-03] Irmansyah, Faried. 2003. "Pengantar Database". Akses dari: <http://ikc.cbn.net.id/umum/faried-database.php>  
Tanggal Akses: 10 September 2005.
- [MAR-03] Marcus, Teddy. 2003 "Pemrograman Delphi untuk Pemula: IDE dan Struktur Pemrograman". Akses dari: <http://ikc.cbn.net.id/umum/teddy-delphi.php>  
Tanggal Akses: 10 September 2005.
- [NIM-00] Nimran, Umar. 2000. "Buku Pedoman Pendidikan Universitas Brawijaya". Malang: Universitas Brawijaya.
- [PIN-05] Pinandito, Aryo. 2005. "Pemrograman Aplikasi Flash Yang Standalone Dan Ekstensibel Pada Katalog Buku Perpustakaan Dengan Menggunakan Relational Database Management System (RDBMS) MySQL". Skripsi tidak diterbitkan. Malang: Jurusan Teknik Elektro Universitas Brawijaya.
- [POH-97] Pohan, Husni Iskandar. 1997. "Pengantar Perancangan Sistem". Jakarta: Erlangga.
- [POW-04] Power Designer Tutorial. 2004. "Sybase® PowerDesigner® Conceptual Data Model. Getting Started". Sybase, Inc.

- [POE-04] Power Designer Tutorial. 2004. "Sybase® PowerDesigner® Physical Data Model. Getting Started". Sybase, Inc.
- [PRA-00] Pranata, Antony. 2000. "Pemrograman Borland Delphi". Yogyakarta: Andi.
- [WAL-00] Waljiyanto. 2000 "Sistem Basis Data: Analisis dan Pemodelan". Yogyakarta: J & J Learning.



**LAMPIRAN A : Generate Database Sybase Power Designer 10***Hasil Generate Database pada Sybase Power Designer 10.*

```
Checking package ...
- Circular references
- Constraint name uniqueness
- Constraint name maximum length
- Circular dependencies
Checking table ...
- Table name uniqueness
- Table code uniqueness
- Table name maximum length
- Table code maximum length
- Existence of column
- Existence of index
- Existence of key
- Number of auto-incremented columns
- Existence of reference
- Table mapping not defined
- Column mapping not defined
Checking column of table
- Column name uniqueness
- Column code uniqueness
- Column code maximum length
- Domain divergence
- Column mandatory
- Detect inconsistencies between check parameters
- Precision > Maximum Length
- Undefined data type
- Foreign key column data type divergence
- Foreign key column constraint divergence
- Column with sequence not in a key
- Auto-incremented column with data type not numeric
- Missing computed column expression
Checking index of table
- Index name uniqueness
- Index code uniqueness
- Index code maximum length
- Existence of index column
- Index column count
- Index inclusion
Checking key of table
- Key name uniqueness
- Key code uniqueness
- Key code maximum length
- Existence of column
- Key inclusion
- Multi-column key has sequence column
Checking reference ...
- Reference name uniqueness
- Reference code uniqueness
- Reflexive and mandatory reference
- Existence of reference join
- Reference code maximum length
- Incomplete join
- Join order
```

```
0 error(s), 0 warning(s).
The Physical Data Model is correct, no errors were found.
```

```
Database Generation
Generation: Check model starting...
Generation: Check model successful.
Sorting objects...
Sort completed.
Script Generation...
Creating Tables...
-> Table: d_hadir (D_HADIR)
-> Table: d_jadwal (D_JADWAL)
-> Table: d_peserta_mk (D_PESERTA_MK)
-> Table: m_dosen (M_DOSEN)
-> Table: m_hari (M_HARI)
-> Table: m_krk (M_KRK)
-> Table: m_login (M_LOGIN)
-> Table: m_mhs (M_MHS)
-> Table: m_mk (M_MK)
Creating References...
-> Reference: FK_D_JADWAL_D_HADIR (FK_D_JADWAL_D_HADIR)
-> Reference: FK_M_MHS_D_HADIR (FK_M_MHS_D_HADIR)
-> Reference: FK_M_DOSEN_D_JADWAL (FK_M_DOSEN_D_JADWAL)
-> Reference: FK_M_HARI_D_JADWAL (FK_M_HARI_D_JADWAL)
-> Reference: FK_M_KRK_D_JADWAL (FK_M_KRK_D_JADWAL)
-> Reference: FK_M_MK_D_JADWAL (FK_M_MK_D_JADWAL)
-> Reference: FK_M_MHS_D_PESERTA_MK (FK_M_MHS_D_PESERTA_MK)
-> Reference: FK_M_LOGIN_M_DOSEN (FK_M_LOGIN_M_DOSEN)
Script Generation completed
Generation successful
```



LAMPIRAN B : Z-3080 barcode scanner specifications

# Z-3080 CCD BARCODE SCANNER

## SPECIFICATIONS

### OPTICAL

<b>Optical System</b>	2048 pixel CCD (charge-coupled device)
<b>Resolution</b>	0.127mm (5 mils) (CODE 39, PCS=90%, when contact)
<b>Scanning Width</b>	80mm (3.15")
<b>PCS Value</b>	30% or more
<b>Light Source</b>	Visible LED (wave length 660nm)
<b>Scan Rate</b>	100 scans/sec
<b>Reading Distance</b>	20mm (EAN = 100%, PCS = 90%)

### ELECTRICAL

<b>Supply Voltage</b>	5VDC ± 10%
<b>Operating Current</b>	140 mA (Standard Version) 10 μA (Special Version) (Zero Power Version, before press trigger)

### ENVIRONMENTAL

<b>Operating Temperature</b>	0 ~ 50°C
<b>Storage Temperature</b>	-10 ~ 70°C
<b>Humidity</b>	5 ~ 95% RH (non-condensing)
<b>Shock</b>	1.0 meter drop to concrete
<b>Max Ambient Light Rejection</b>	1500 Lux (Fluorescence) 800 Lux (Sunlight)

### MECHANICAL

<b>Cable Length</b>	Straight (2m)
<b>Connector Type</b>	RJ-45 phone jack connector
<b>Weight</b>	170g (5.9oz) excluding cable and connector
<b>Case Material</b>	ABS plastic

### MULTI-INTERFACE

Keyboard emulation, C-MOS Serial, RS-232C Serial, and Wand emulation. All in one unit. Optional USB interface.

### KEYBOARD EMULATIONS

IBM PC/XT/AT, IBM PS/2 Model 30-80, IBM 5550, 5530-SC, 5530-ZC, IBM 3196, 3472/3477, 5295 Terminal, NEC 9801, Apple MAC II, and more ...

### DECODING CAPABILITIES

UPC A/E, EAN-8/JAN-8, EAN-13/JAN-13, Full ASCII Code 39, Code 39, Interleaved 2 of 5, Code 93, Codabar, Code 128, Chinese postcode, IATA, EAN-128, MSI Plessey  
Optional: Code 11, STD 2 of 5, & Matrix 2 of 5

### PROGRAMMABLE CHARACTERISTICS

#### Built-in decoder:

- Type of interface
- Code type selection, check digit selection
- Decoding option
- Transmitted character delay
- Header selection, trailer selection
- Good read beep tone and volume
- Barcode identifier

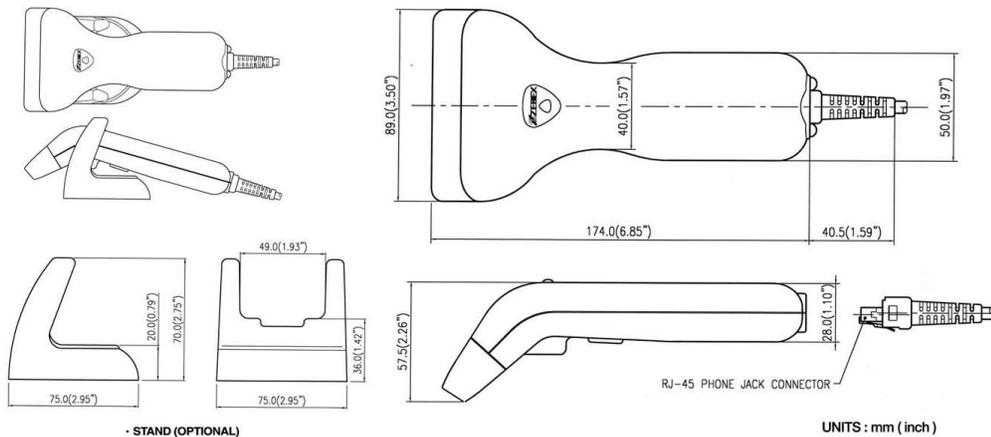
#### Keyboard Emulation Configuration:

Inter-message delay, keyboard type and keyboard language

#### Serial Interface Type:

ACK/NAK, Xon/Xoff, RTS/CTS, Raud Rate, Data Bits, Stop Bits, Parity

Specifications are subject to change without notice. All brand names and registered trademarks are the property of their owners.  
The word ZEBEX and the ZEBEX logo are registered trademarks of Zebex Industries Inc.  
© 2003 ZEBEX Catalogue No. CAT-30122-03



- STAND (OPTIONAL)

UNITS : mm ( inch )



HEADQUARTERS  
**ZEBEX INDUSTRIES INC.**  
2F, 531-1, Chung Cheng Road,  
Hsin-Tien City, Taipei, Taiwan  
Telephone / +886-2-2218-2018  
Facsimile / +886-2-2218-8670  
WebSite / www.zebex.com.tw  
E-mail / mail@zebex.com.tw

USA  
**ZEBEX AMERICA INC.**  
11 Ilene Court, Unit #2,  
Hillsborough, NJ 08844  
Telephone / +1-908-359-2070  
Facsimile / +1-908-359-1272  
WebSite / www.zebex.com  
E-mail / sales@zebex.com

CHINA  
**ZEBEX INDUSTRIES (SHANGHAI) INC.**  
Number 1998, Qixin Road,  
Minhang District, Shanghai, China  
Telephone / +86-21-5479-4420  
Facsimile / +86-21-5479-4421  
WebSite / www.zebex.com.tw  
E-mail / zebex@21cn.net