BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap perangkat lunak Aplikasi Monitoring Penjadwalan Kelas dan Berita Acara Perkuliahan(BAP) Berbasis RFID di Universitas Komputer Indonesia, yaitu tahap penerjemahan kebutuhan pembangunan aplikasi kedalam representasi kebutuhan perangkat lunak. Setelah implementasi maka dilakukan pengujian sistem yang dibangun untuk melihat kekurangan-kekurangan yang ada pada aplikasi.

4.1 Implementasi Sistem

Setelah aplikasi dianalisis dan dirancang secara rinci, tahap selanjutnya adalah implementasi. Tujuan dari implementasi adalah untuk mengkonfirmasi modul-modul perancangan sehingga pengguna dapat memberikan masukan kepada pengembangan sistem.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut :

1. *Processor* : AMD Phenom II X2 565 3,4 GHz
2. *Harddisk* : 320 GB
3. RAM DDR3 8 GB PC10600
4. Monitor : LCD Resolusi 1366 x 768
5. RFID *Reader* : RFID ID-12
6. RFID *tag* : EM4001
7. Serial Port *on-board* : COM1
8. *Keyboard*
9. *Mouse*

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun dan mengimplementasikan sistem adalah sebagai berikut :

1. Sistem Operasi Windows 7 Professional
2. MySQL 5.5.16
3. WampServer 2.2
4. Microsoft .NET Framework 4.0

4.1.3 Implementasi Basis Data

Implementasi basis data diambil berdasarkan perancangan basis data yang dibuat sebelumnya. Secara fisik, implementasi basis data diimplementasikan menggunakan perangkat lunak DBMS MySQL. Berikut adalah *query* sql untuk setiap tabel.

1. Tabel Absensi

Query sql untuk tabel Absensi adalah sebagai berikut :

```
CREATE TABLE IF NOT EXISTS `absensi` (
  `kodekelasperkuliahan` int(11) NOT NULL,
  `nim` varchar(8) NOT NULL,
  `m1` tinyint(4) DEFAULT '0',
  `m2` tinyint(4) DEFAULT '0',
  `m3` tinyint(4) DEFAULT '0',
  `m4` tinyint(4) DEFAULT '0',
  `m5` tinyint(4) DEFAULT '0',
  `m6` tinyint(4) DEFAULT '0',
  `m7` tinyint(4) DEFAULT '0',
  `m8` tinyint(4) DEFAULT '0',
  `m9` tinyint(4) DEFAULT '0',
  `m10` tinyint(4) DEFAULT '0',
  `m11` tinyint(4) DEFAULT '0',
  `m12` tinyint(4) DEFAULT '0',
  `m13` tinyint(4) DEFAULT '0',
  `m14` tinyint(4) DEFAULT '0',
  `m15` tinyint(4) DEFAULT '0',
  `m16` tinyint(4) DEFAULT '0',
  PRIMARY KEY (`kodekelasperkuliahan`,`nim`),
  KEY `nim` (`nim`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `absensi`
  ADD CONSTRAINT `absensi_ibfk_2` FOREIGN KEY (`nim`)
REFERENCES `mahasiswa` (`NIM`) ON UPDATE CASCADE,
  ADD CONSTRAINT `absensi_ibfk_1` FOREIGN KEY
(`kodekelasperkuliahan`) REFERENCES `kelas_perkuliahan`
(`KodeKelasPerkuliahan`) ON UPDATE CASCADE;
```

2. Tabel bap

Query sql untuk tabel bap adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `bap` (
  `KodeKelasPerkuliahan` int(11) NOT NULL,
  `PertemuanKe` tinyint(4) NOT NULL,
  `WaktuPengisian` datetime NOT NULL,
  `Materi` text NOT NULL COMMENT 'materi yang dibahas',
  `JenisPerkuliahan` enum('Perkuliahan Normal','Kuliah
Pengganti') NOT NULL,
  `KodeLog` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`KodeKelasPerkuliahan`,`PertemuanKe`),
  UNIQUE KEY `WaktuPengisian` (`WaktuPengisian`),
  UNIQUE KEY `idxKodeLogRuang` (`KodeLog`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `bap`
  ADD CONSTRAINT `bap_ibfk_1` FOREIGN KEY
(`KodeKelasPerkuliahan`) REFERENCES `kelas_perkuliahan`
(`KodeKelasPerkuliahan`) ON UPDATE CASCADE,
  ADD CONSTRAINT `bap_ibfk_2` FOREIGN KEY (`KodeLog`)
REFERENCES `log_ruang` (`KodeLog`) ON UPDATE CASCADE;
```

3. Tabel dosen

Query sql untuk tabel dosen adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `dosen` (
  `KodeDosen` int(11) NOT NULL AUTO_INCREMENT,
  `KodeDosens` char(3) NOT NULL DEFAULT '',
  `NIP` varchar(11) DEFAULT NULL,
  `NAMA` varchar(30) DEFAULT NULL,
  `GLR1` varchar(20) DEFAULT NULL,
  `GLR2` varchar(20) DEFAULT NULL,
  `TMPLHR` varchar(15) DEFAULT NULL,
  `TGLLHR` date DEFAULT NULL,
  `JenisKelamin` char(1) DEFAULT NULL,
  `AGAMA` char(1) DEFAULT NULL,
  `ALAMAT` text,
  `TELP` varchar(40) DEFAULT NULL,
  `KodeJurusan` char(3) DEFAULT NULL,
  `Pass` varchar(41) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL,
  `NIDN` varchar(20) DEFAULT NULL,
```

```
  `Status` enum('Aktif','Tidak Aktif','Cuti','Keluar')
DEFAULT 'Aktif',
  `Jenis` enum('Dosen Tetap','Dosen Tetap Khusus','Dosen
Luar Biasa') DEFAULT NULL,
  PRIMARY KEY (`KodeDosen`),
  UNIQUE KEY `IdxNIDN` (`NIDN`),
  UNIQUE KEY `IdxEmail` (`Email`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=885
;
```

4. Tabel fakultas

Query sql untuk tabel fakultas adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `fakultas` (
  `KodeFakultas` varchar(2) NOT NULL,
  `Nama` varchar(50) DEFAULT NULL,
  `KodeDekan` varchar(3) DEFAULT NULL,
  `KodeFak_EPSBED` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`KodeFakultas`),
  KEY `K1` (`KodeFakultas`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `fakultas`
  ADD CONSTRAINT `fakultas_ibfk_1` FOREIGN KEY (`KodeDekan`)
REFERENCES `dosen` (`KodeDosen`) ON UPDATE CASCADE;
```

5. Tabel frs

Query sql untuk tabel frs adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `frs` (
  `semester` varchar(5) NOT NULL,
  `nim` varchar(8) NOT NULL,
  `kodematakuliah` varchar(10) NOT NULL,
  `kurikulum` varchar(4) NOT NULL,
  `kodekelas` varchar(7) DEFAULT NULL COMMENT 'tidak
digunakan',
  `KodeKelasPerkuliahan` int(11) DEFAULT NULL,
  `waktu_isi` datetime DEFAULT NULL,
  `disetujui` enum('T','Y') DEFAULT 'T',
  `waktu_setuju` datetime DEFAULT NULL,
  `jenis_perwalian` enum('FRS','PRS') DEFAULT 'FRS',
  PRIMARY KEY
(`semester`,`nim`,`kodematakuliah`,`kurikulum`),
  KEY `IdxKelasPerkuliahan` (`KodeKelasPerkuliahan`),
  KEY `idxsemnim` (`semester`,`nim`),
  KEY `fk_frs_mahasiswa1` (`nim`),
  KEY `fk_frs_matakuliah1` (`kodematakuliah`),
  KEY `kurikulum` (`kurikulum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `frs`
  ADD CONSTRAINT `frs_ibfk_1` FOREIGN KEY (`nim`) REFERENCES
`mahasiswa` (`NIM`) ON UPDATE CASCADE,
  ADD CONSTRAINT `frs_ibfk_2` FOREIGN KEY (`kodematakuliah`)
REFERENCES `matakuliah` (`KodeMataKuliah`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `frs_ibfk_3` FOREIGN KEY (`kurikulum`)
REFERENCES `matakuliah` (`KURIKULUM`) ON UPDATE CASCADE;
```

6. Tabel jurusan

   Query sql untuk tabel jurusan adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `jurusan` (
  `KodeJurusan` varchar(3) NOT NULL,
  `Nama` varchar(47) DEFAULT NULL,
  `KodeFakultas` varchar(2) DEFAULT NULL,
  `KodeDosenKajur` int(11) DEFAULT NULL,
  `KodeDosenSekjur` int(11) DEFAULT NULL,
  `Pass` varchar(50) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`KodeJurusan`),
  KEY `FK_jurusan_fakultas` (`KodeFakultas`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `jurusan`
  ADD CONSTRAINT `jurusan_ibfk_1` FOREIGN KEY
(`KodeFakultas`) REFERENCES `fakultas` (`KodeFakultas`) ON
UPDATE CASCADE;
```

7. Tabel kelas

   Query sql untuk tabel kelas adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `kelas` (
  `KodeKelas` varchar(10) NOT NULL,
  `KodeProdi` varchar(3) DEFAULT NULL,
  `Nama` varchar(20) DEFAULT NULL COMMENT 'Nama Kelas',
  `Tahun` varchar(4) DEFAULT NULL COMMENT 'Tahun Masuk',
  `KodeJenisKelas` varchar(1) DEFAULT NULL COMMENT 'Jenis
Kelas (R)eguler/(K) Karyawan',
  `KodeDosenWali` int(11) DEFAULT NULL,
  PRIMARY KEY (`KodeKelas`),
  KEY `FK_kelas_prodi` (`KodeProdi`),
  KEY `FK_kelas_jeniskelas` (`KodeJenisKelas`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `kelas`
  ADD CONSTRAINT `kelas_ibfk_1` FOREIGN KEY (`KodeProdi`)
REFERENCES `prodi` (`KodeProdi`) ON UPDATE CASCADE;
```

8. Tabel kelas_perkuliahan

Query sql untuk tabel kelas_perkuliahan adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `kelas_perkuliahan` (
  `KodeKelasPerkuliahan` int(11) NOT NULL AUTO_INCREMENT,
  `Semester` varchar(5) NOT NULL,
  `KodeKelas` varchar(7) DEFAULT '',
  `KodeMatakuliah` varchar(10) NOT NULL,
  `Kurikulum` varchar(4) DEFAULT NULL,
  `KodeHari` tinyint(1) DEFAULT NULL,
  `KodeWaktuAwal` smallint(6) DEFAULT NULL,
  `KodeWaktuAkhir` smallint(6) DEFAULT NULL,
  `KodeRuang` varchar(15) DEFAULT NULL,
  `SubKelas` char(1) DEFAULT 'A',
  `NamaKelasAlt` varchar(50) DEFAULT NULL COMMENT 'Nama
Kelas alternatif',
  `KodeDosenLama` varchar(3) DEFAULT NULL,
  `MaksimalMhs` int(11) DEFAULT '0',
  PRIMARY KEY (`KodeKelasPerkuliahan`),
  KEY `idxKelas` (`KodeKelas`),
  KEY `idxKurikulum` (`Kurikulum`),
  KEY `idxMatakuliah` (`KodeMatakuliah`),
  KEY `idxSemester` (`Semester`),
  KEY `idxRuang` (`KodeRuang`),
  KEY `W1` (`KodeWaktuAwal`),
  KEY `W2` (`KodeWaktuAkhir`),
  KEY `idxHari` (`KodeHari`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=13351
;

ALTER TABLE `kelas_perkuliahan`
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_1` FOREIGN KEY
(`Semester`) REFERENCES `frs` (`semester`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_2` FOREIGN KEY
(`KodeKelas`) REFERENCES `kelas` (`KodeKelas`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_3` FOREIGN KEY
(`Kurikulum`) REFERENCES `matakuliah` (`KURIKULUM`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_4` FOREIGN KEY
(`KodeRuang`) REFERENCES `ruang` (`KodeRuang`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_5` FOREIGN KEY
(`KodeWaktuAwal`) REFERENCES `tref_waktu` (`KodeWaktu`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_6` FOREIGN KEY
(`KodeWaktuAkhir`) REFERENCES `tref_waktu` (`KodeWaktu`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_ibfk_7` FOREIGN KEY
(`KodeHari`) REFERENCES `tref_hari` (`KodeHari`) ON UPDATE
CASCADE;
```

9. Tabel kelas_perkuliahan_dosen

Query sql untuk tabel kelas_perkuliahan_dosen adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `kelas_perkuliahan_dosen` (
  `kodekelasperkuliahan` int(11) NOT NULL DEFAULT '0',
  `kodedosen` int(11) DEFAULT NULL,
  `banyakpertemuan` int(11) DEFAULT NULL,
  UNIQUE KEY `idx_kelasperkuliahan_dosen`
(`kodekelasperkuliahan`,`kodedosen`),
  KEY `idxKodeDosen` (`kodedosen`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `kelas_perkuliahan_dosen`
  ADD CONSTRAINT `kelas_perkuliahan_dosen_ibfk_2` FOREIGN
KEY (`kodedosen`) REFERENCES `dosen` (`KodeDosen`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `kelas_perkuliahan_dosen_ibfk_1` FOREIGN
KEY (`kodekelasperkuliahan`) REFERENCES `kelas_perkuliahan`
(`KodeKelasPerkuliahan`) ON UPDATE CASCADE;
```

10. Tabel log_ruang

Query sql untuk tabel log_ruang adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `log_ruang` (
  `KodeLog` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `KodeRuang` varchar(20) NOT NULL,
  `tglAkses` date NOT NULL,
  `JamAksesMasuk` time NOT NULL,
  `JamAksesKeluar` time DEFAULT NULL,
  `KodePengguna` int(11) NOT NULL,
  `JenisPengguna` varchar(20) DEFAULT NULL,
  `KodeKelasPerkuliahan` int(11) DEFAULT NULL,
  `Keterangan` varchar(50) DEFAULT NULL,
  `StatusPintu` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`KodeLog`),
  KEY `idx_kelas_perkuliahan` (`KodeKelasPerkuliahan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

ALTER TABLE `log_ruang`
  ADD CONSTRAINT `log_ruang_ibfk_1` FOREIGN KEY
(`KodeKelasPerkuliahan`) REFERENCES `kelas_perkuliahan`
(`KodeKelasPerkuliahan`) ON UPDATE CASCADE;
```

11. Tabel mahasiswa

Query SQL untuk tabel mahasiswa adalah sebagai berikut :

```
CREATE TABLE IF NOT EXISTS `mahasiswa` (
  `NIM` varchar(8) NOT NULL DEFAULT '',
  `Nama` varchar(100) NOT NULL DEFAULT '',
  `Tahun` varchar(4) NOT NULL DEFAULT '',
  `Jalur` char(1) NOT NULL DEFAULT '',
```

```
  `Status` char(1) NOT NULL DEFAULT '' COMMENT 'Aktif (A)
atau Cuti (C)',
  `KodeKelas` varchar(7) DEFAULT NULL,
  `Jenis` char(1) DEFAULT NULL,
  `Pass` varchar(50) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL,
  `kddsn` int(11) DEFAULT NULL,
  `SemesterMasuk` varchar(5) DEFAULT NULL,
  `Kekhususan` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`NIM`),
  UNIQUE KEY `IdxEMail` (`Email`),
  KEY `idxKelas` (`KodeKelas`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `mahasiswa`
  ADD CONSTRAINT `mahasiswa_ibfk_1` FOREIGN KEY
(`KodeKelas`) REFERENCES `kelas` (`KodeKelas`) ON UPDATE
CASCADE;
```

12. Tabel matakuliah

Query sql untuk tabel matakuliah adalah sebagai berikut :

```
CREATE TABLE IF NOT EXISTS `matakuliah` (
  `KodeProdi` varchar(3) DEFAULT NULL,
  `KodeMataKuliah` varchar(10) NOT NULL DEFAULT '',
  `MATAKULIAH` varchar(100) DEFAULT NULL,
  `SKS` tinyint(1) unsigned DEFAULT NULL,
  `SEMESTER` tinyint(1) unsigned DEFAULT NULL,
  `PROGRAM` char(2) NOT NULL DEFAULT 'S1',
  `WP` char(1) NOT NULL DEFAULT 'W',
  `ALIAS` varchar(30) DEFAULT NULL,
  `TLG` char(1) NOT NULL DEFAULT 'T',
  `KURIKULUM` varchar(4) NOT NULL DEFAULT '',
  `KETERANGAN` varchar(30) DEFAULT NULL,
  `KodeKelompokMK` varchar(1) DEFAULT NULL,
  `Disurvey` enum('Y','T') DEFAULT 'Y',
  `Kekhususan` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`KodeMataKuliah`,`KURIKULUM`),
  KEY `KURIKULUM` (`KURIKULUM`),
  KEY `MATAKULIAH` (`MATAKULIAH`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Tabel prodi

Query sql untuk tabel prodi adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `prodi` (
  `KodeProdi` varchar(3) NOT NULL,
  `KodeJurusan` varchar(3) DEFAULT NULL,
  `KDPRODI_LAMA` varchar(2) DEFAULT NULL,
  `KodeFakultas` varchar(2) DEFAULT NULL,
  `KodeJenjang` varchar(2) DEFAULT NULL,
  `KodeProdi_EPSBED` varchar(5) DEFAULT NULL,
```

```
  `Nama` varchar(40) DEFAULT NULL,
  `TahunKurikulum` varchar(4) DEFAULT NULL,
  `Dibuka` varchar(1) DEFAULT 'Y',
  `TGAWLMSPST` datetime DEFAULT NULL,
  `SMAWLMSPST` varchar(5) DEFAULT NULL,
  `STATUMSPST` varchar(1) DEFAULT NULL,
  `MLSEMMSPST` varchar(5) DEFAULT NULL,
  `SKSTTMSPST` smallint(5) DEFAULT NULL,
  `EMAILMSPST` varchar(40) DEFAULT NULL,
  `NOMSKMSPST` varchar(40) DEFAULT NULL,
  `TGLSKMSPST` datetime DEFAULT NULL,
  `TGLAKMSPST` datetime DEFAULT NULL,
  `NOMBAMSPST` varchar(40) DEFAULT NULL,
  `TGLBAMSPST` datetime DEFAULT NULL,
  `TGLABMSPST` datetime DEFAULT NULL,
  `KDSTAMSPST` varchar(1) DEFAULT NULL,
  `KDFREMSPST` varchar(1) DEFAULT NULL,
  `KDPELMSPST` varchar(1) DEFAULT NULL,
  `NOKPSMSPST` varchar(10) DEFAULT NULL,
  `TELPSMSPST` varchar(20) DEFAULT NULL,
  `TELPOMSPST` varchar(20) DEFAULT NULL,
  `FAKSIMSPST` varchar(20) DEFAULT NULL,
  `NMOPRMSPST` varchar(40) DEFAULT NULL,
  `TELPRMSPST` varchar(20) DEFAULT NULL,
  `LISAHMSPST` varchar(1) DEFAULT NULL,
  `KDPS1MSPST` varchar(5) DEFAULT NULL,
  `Singkatan` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`KodeProdi`),
  UNIQUE KEY `KDPRODI` (`KDPRODI_LAMA`),
  KEY `K1` (`KodeJenjang`,`KodeProdi_EPSBED`),
  KEY `K2`
(`KodeFakultas`,`KodeJenjang`,`KodeProdi_EPSBED`),
  KEY `FK_prodi_jurusan` (`KodeJurusan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `prodi`
  ADD CONSTRAINT `prodi_ibfk_2` FOREIGN KEY
(`KodeFakultas`) REFERENCES `fakultas` (`KodeFakultas`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `prodi_ibfk_1` FOREIGN KEY
(`KodeJurusan`) REFERENCES `jurusan` (`KodeJurusan`) ON
UPDATE CASCADE;
```

14. Tabel rfid_dosen

Query sql untuk tabel rfid_dosen adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `rfid_dosen` (
  `kodeRFID` varchar(20) CHARACTER SET utf8 NOT NULL,
  `KodeDosen` int(11) NOT NULL,
  PRIMARY KEY (`kodeRFID`),
  UNIQUE KEY `KodeDosen` (`KodeDosen`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `rfid_dosen`
  ADD CONSTRAINT `rfid_dosen_ibfk_1` FOREIGN KEY
(`KodeDosen`) REFERENCES `dosen` (`KodeDosen`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

15. Tabel rfid_mahasiswa

Query sql untuk tabelrfid_mahasiswa adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `rfid_mahasiswa` (
  `kodeRFID` varchar(20) CHARACTER SET utf8 NOT NULL,
  `NIM` varchar(8) NOT NULL,
  PRIMARY KEY (`kodeRFID`),
  UNIQUE KEY `NIM` (`NIM`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `rfid_mahasiswa`
  ADD CONSTRAINT `rfid_mahasiswa_ibfk_1` FOREIGN KEY
(`NIM`) REFERENCES `mahasiswa` (`NIM`) ON DELETE CASCADE
ON UPDATE CASCADE;
```

16. Tabel romawi

Query sql untuk tabel romawi adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `romawi` (
  `kodeRomawi` int(5) NOT NULL AUTO_INCREMENT,
  `Nama` varchar(5) NOT NULL,
  PRIMARY KEY (`kodeRomawi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=17
;
```

17. Tabel ruang

Query sql untuk tabel ruang adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `ruang` (
  `KodeRuang` varchar(20) NOT NULL DEFAULT '',
  `JenisPerkuliahan` varchar(1) DEFAULT NULL,
  PRIMARY KEY (`KodeRuang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

18. Tabel sekretariat_jurusan

Query sql untuk tabel ruang adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `sekretariat_jurusan` (
  `NIP` varchar(11) NOT NULL,
  `Pass` varchar(41) NOT NULL,
  `Nama` varchar(30) NOT NULL,
  `KodeJurusan` char(3) NOT NULL,
```

```
  PRIMARY KEY (`NIP`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

19. Tabel tref_hari

```
CREATE TABLE IF NOT EXISTS `tref_hari` (
  `KodeHari` tinyint(1) NOT NULL DEFAULT '0',
  `Nama` varchar(6) DEFAULT NULL,
  PRIMARY KEY (`KodeHari`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

20. Tabel tref_waktu

```
CREATE TABLE IF NOT EXISTS `tref_waktu` (
  `KodeWaktu` smallint(6) NOT NULL DEFAULT '0',
  `Awal` varchar(5) DEFAULT NULL,
  `Akhir` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`KodeWaktu`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.1.4 Implementasi Antar muka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun. Berikut ini adalah implementasi antarmuka dari perangkat lunak Aplikasi Monitoring Penjadwalan Kelas dan Berita Acara Perkuliahan(BAP) Berbasis RFID di Universitas Komputer Indonesia..

1. Antar muka Aplikasi Monitoring Kelas

Merupakan tampilan utama dari aplikasi monitoring kelas yang berfungsi membuka dan mengunci pintu ruangan.



Gambar 4.1 Antar muka Aplikasi Monitoring Kelas

2. Antar muka Aplikasi Pengisian Berita Acara Perkuliahan

a) Antar muka Login Dosen

Merupakan tampilan login bagi dosen untuk verifikasi data sebelum mengisi berita acara perkuliahan.



Gambar 4.2 Antar muka Login Dosen

b) Antar muka Menu Utama Dosen

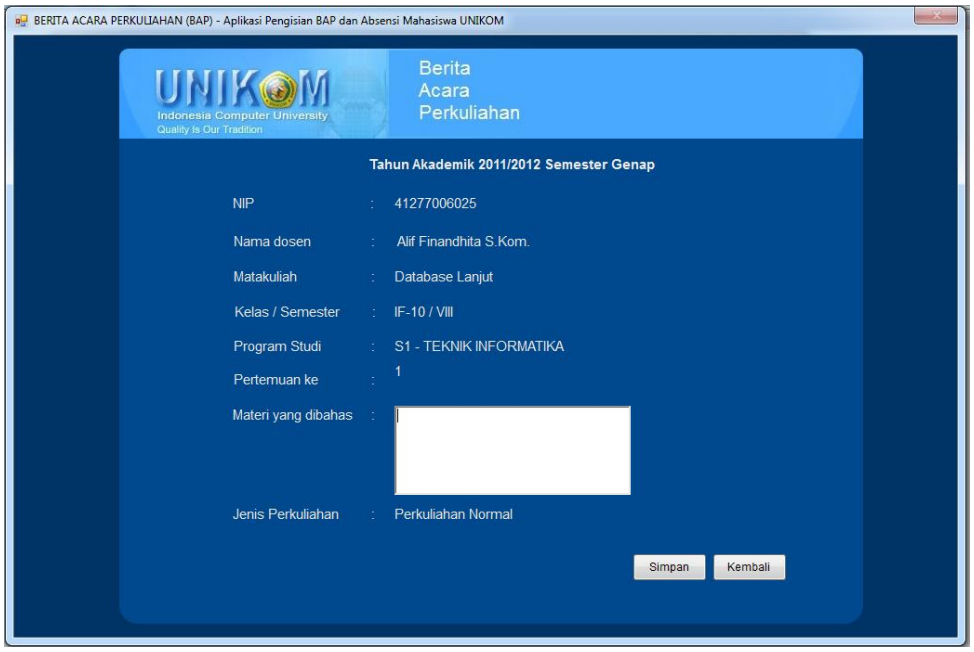
Merupakan tampilan menu yang tampil setelah login dosen untuk pengisian BAP



Gambar 4.3 Antar muka Menu Utama

c) Antar muka Halaman Pengisian Berita Acara Perkuliahan

Merupakan form untuk mengisi berita acara perkuliahan (BAP). Disini dosen cukup mengisi materi saja.



Gambar 4.4 Antar muka form pengisian BAP

3. Aplikasi Pencetakan Laporan Berita Acara Perkuliahan

a) Antar muka Login Sekretariat Jurusan

Merupakan tampilan login bagi sekretariat jurusan untuk dapat mencetak Berita Acara Perkuliahan



Gambar 4.5 Antar muka Absensi Kehadiran

b) Antar muka Menu Utama Sekretariat Jurusan

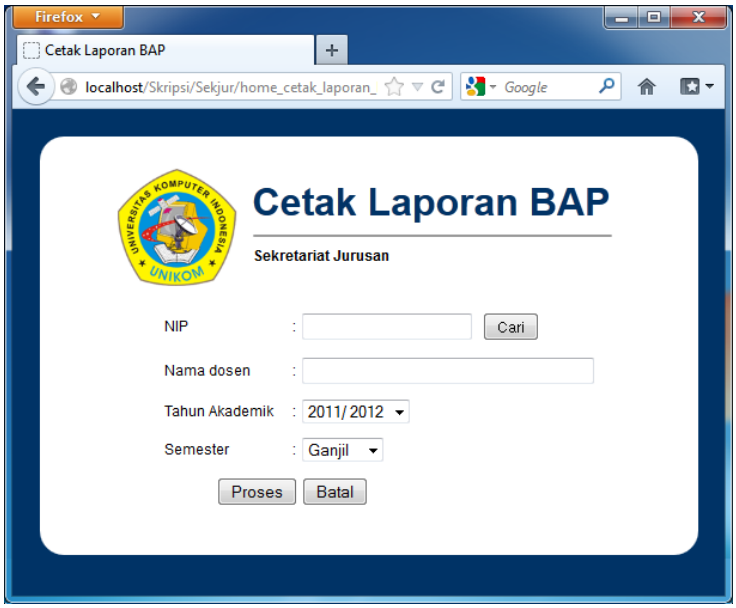
Merupakan form yang ditampilkan setelah staf sekretariat jurusan berhasil login.



Gambar 4.6 Antar muka Menu Utama Sekretariat Jurusan

c) Antar muka Pencetakan Berita Acara Perkuliahan

Merupakan tampilan untuk mencari data dosen yang akan dicetak BAP-nya.



Gambar 4.7 Antar muka Pencetakan Berita Acara Perkuliahan

d) Antar muka Laporan Berita Acara perkuliahan

Tampilan laporan Berita Acara Perkuliahan dosen yang siap cetak.



No	Nama Mata Kuliah	SKS	SMT	Kelas / Program	Jurusan	Waktu Mengajar		Ruang	Materi yang Dibahas	Tanggal Pengisian	Jenis Perkuliahan
						Hari	Jam				
1	Database Lanjut	3	VIII	IF-3 / S1	TEKNIK INFORMATIKA	Senin	13.00 - 15.15	LAB-5			
2	Database Lanjut	3	VIII	IF-1 / S1	TEKNIK INFORMATIKA	Senin	15.15 - 17.30	LAB-5			
3	Database Lanjut	3	VIII	IF-18 / S1	TEKNIK INFORMATIKA	Senin	17.30 - 19.45	LAB-5			
4	Database Lanjut	3	VIII	IF-9 / S1	TEKNIK INFORMATIKA	Selasa	13.00 - 15.15	LAB-5			
5	Database Lanjut	3	VIII	IF-7 / S1	TEKNIK INFORMATIKA	Selasa	15.15 - 17.30	LAB-5			
6	Database Lanjut	3	VIII	IF-2 / S1	TEKNIK INFORMATIKA	Rabu	07.00 - 09.15	LAB-5			
7	Database Lanjut	3	VIII	IF-4 / S1	TEKNIK INFORMATIKA	Rabu	13.45 - 16.00	LAB-5			
8	Database Lanjut	3	VIII	IF-6 / S1	TEKNIK INFORMATIKA	Rabu	16.00 - 18.15	LAB-5			
9	Database Lanjut	3	VIII	IF-8 / S1	TEKNIK INFORMATIKA	Kamis	07.00 - 09.15	LAB-5			
10	Database Lanjut	3	VIII	IF-5 / S1	TEKNIK INFORMATIKA	Kamis	13.45 - 16.00	LAB-5			
11	Database Lanjut	3	VIII	IF-13 / S1	TEKNIK INFORMATIKA	Kamis	16.00 - 18.15	LAB-5			

Gambar 4.8 Antar muka Laporan Berita Acara perkuliahan

4. Antar muka Aplikasi Kuliah Pengganti

a. Antar muka Login dosen

Merupakan tampilan form login dosen untuk masuk ke halaman utama aplikasi kuliah pengganti



Gambar 4.9 Antar muka Login Dosen Aplikasi Kuliah Pengganti

b. Antar muka halaman home



Gambar 4.10 Antar muka Halaman Home Aplikasi Kuliah Pengganti

4.2 Pengujian

Pengujian perangkat lunak adalah proses menelusuri dan mempelajari sebuah program dalam rangka menemukan kesalahan pada perangkat lunak sebelum diserahkan kepada *end user*.

Pengujian perangkat lunak ini menggunakan pengujian *black box.* Pengujian *black box* merupakan pengujian yang digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Tujuan dari metode *black box* ini adalah untuk menemukan kesalahan fungsi pada program.

4.2.1 Pengujian Alpha

Pengujian alpha merupakan pengujian yang menitik beratkan pada hasil output dari kendali input yang dimasukan pada tampilan kendali input tampilan. Pengujian dapat dikatakan berhasil apabila output sesuai dengan kendali input yang dimasukan pada setiap tampilan.

4.2.1.1 Skenario Pengujian Alpha

Skenario pengujian alpa menjelaskan skenario pengujian sistem, yakni sebagai berikut :

Tabel 4.1 Skenario pengujian aplikasi monitoring ruang kelas

Kelas Uji	Butir Uji	Jenis Pengujian
Membuka pintu	Membuka pintu	*Black box*
Mengunci pintu	Mengunci pintu	*Black box*

Tabel 4.2 Skenario pengujian aplikasi pengisian BAP

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi username	*Black box*
	Verifikasi password	*Black box*

Mengisi Berita Acara Perkuliahan	Menampilkan form pengisian Berita Acara Perkuliahan	*Black box*
	Mengisi Berita Acara Perkuliahan	*Black box*
	Menyimpan Berita Acara Perkuliahan	*Black box*

Tabel 4.3 Skenario pengujian aplikasi pencetakan laporan BAP

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi username	*Black box*
	Verifikasi password	*Black box*
Menampilkan laporan	Memasukkan nip dosen, tahun, dan semester akademik sebagai parameter untuk menyeleksi data Berita Acara Perkuliahan	*Black box*
	Menampilkan laporan Berita Acara Perkuliahan	*Black box*

Tabel 4.4 Skenario pengujian aplikasi kuliah pengganti

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi username	*Black box*
	Verifikasi password	*Black box*
Mengisi data kuliah pengganti	Menampilkan form pengisian data kuliah pengganti	*Black box*
	Isi form kuliah pengganti	*Black box*
	Menyimpan data kuliah pengganti	*Black box*

4.2.1.2 Butir Pengujian

Aplikasi Monitoring Ruang Kelas

1. Pengujian Membuka Pintu

Tabel 4.5 Pengujian Membuka Pintu

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu terbuka	Dilakukan pada hari jumat, jam 13.00 WIB, di ruangan LAB-5, untuk jadwal mengajar matakuliah Basis Data Lanjut Kelas IF-10 Semester VIII tahun akademik 2011/2012 genap, Sehingga dapat pintu terbuka.	[√] diterima [ ] ditolak
2.	Tag RFID dosen (Andri Heryandi S.T., M.T.) Kode RFID : 78003E9D4992	Pintu terbuka	Dilakukakn pada hari jumat, jam 20.30. Meskipun dilakukan diluar jadwal mengajar namun dosen telah melakukan pengisian data kuliah pengganti, Sehingga pintu terbuka	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Sistem tidak bereaksi	Status pintu tetap seperti semula, yakni terbuka, Sebab proses baca kartu *tag* dilakukan ketika ruangan masih digunakan oleh orang lain	[√] diterima [ ] ditolak

2.	Tag RFID bukan dosen Kode RFID : 170403723C5E	Pintu tidak terbuka	Pintu tidak terbuka. Hal ini berlaku pada semua waktu dan ruangan	[√] diterima [ ] ditolak
3.	Tag RFID dosen (Andri Heryandi S.T., M.T.) Kode RFID : 78003E9D4992	Pintu tidak terbuka	Pintu tidak terbuka. Proses baca kartu *tag* dilakukan ketika terdapat jadwal mengajar dosen lain	[√] diterima [ ] ditolak
4.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu tidak terbuka	Pintu tidak terbuka, karena jumlah pertemuan yang dilakukan sudah lebih dari 14 kali pertemuan	[√] diterima [ ] ditolak
5.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu tidak terbuka	Pintu tidak terbuka, karena dosen sudah selesai melaksanakan perkuliahan tersebut pada hari ini.	[√] diterima [ ] ditolak
6.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu tidak terbuka	Pintu tidak terbuka, karena dosen sudah telambat lebih dari 30 menit	[√] diterima [ ] ditolak

2. Pengujian Mengunci Pintu

Tabel 4.6 Pengujian Mengunci Pintu

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tag RFID dosen (Alif Finandhita	Pintu terkunci	Pintu terkunci. Keadaan : proses	[√] diterima [ ] ditolak

	S.Kom.) Kode RFID : 2B0015AF0293		baca kartu *tag* dilakukan setelah sebelumnya dosen tersebut membuka pintu dan telah mengisi Berita Acara Perkuliahan	
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Sistem tidak bereaksi	Status pintu tetap seperti keadaan semula : terkunci. Sebab, Pintu sudah dalam keadaan terkunci	[√] diterima [ ] ditolak
2.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu tidak terkunci	Pintu tidak terkunci. Penyebab : Pintu dalam keadaan terbuka, namun bukan dibuka oleh dosen yang bersangkutan	[√] diterima [ ] ditolak
3.	Tag RFID dosen (Alif Finandhita S.Kom.) Kode RFID : 2B0015AF0293	Pintu tidak terkunci	Pintu tidak terkunci. Penyebab : Dosen belum mengisi Berita Acara Perkuliahan	[√] diterima [ ] ditolak

Aplikasi Pengisian Berita Acara Perkuliahan

1. Pengujian Login

Tabel 4.7 Pengujian login aplikasi pengisian berita acara perkuliahan

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dan Password dosen, contoh : NIP : 41277006009 Password : irawan	Masuk aplikasi pengisian Berita Acara Perkuliahan	Dapat masuk ke aplikasi pengisian Berita Acara Perkuliahan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dan Password, contoh : NIP : 1 Password : irawan	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
2.	NIP dan Password, contoh : NIP : 41277006009 Password : taryana	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
3.	NIP dan Password, contoh : NIP : 1 Password : 1	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak

2. Pengujian Mengisi Berita Acara Perkuliahan

Tabel 4.8 Pengujian mengisi berita acara perkuliahan

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Materi Berita Acara Perkuliahan : "Pengenalan Oracle"	Berita Acara Perkuliahan tersimpan	Berita Acara Perkuliahan tersimpan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Materi Berita Acara Perkuliahan : (kosong)	Muncul peringatan bahwa materi belum diisi	Muncul peringatan bahwa materi belum diisi	[√] diterima [ ] ditolak
2.	Materi Berita Acara Perkuliahan : "Pengenalan Oracle"	Muncul peringatan "Anda sudah mengisi Berita Acara Perkuliahan !"	Muncul peringatan bahwa Berita Acara Perkuliahan sudah terisi	[√] diterima [ ] ditolak
3.	Materi Berita Acara Perkuliahan : "Pengenalan Oracle"	Muncul peringatan "Jumlah kehadiran mahasiswa tidak boleh nol !"	Muncul peringatan bahwa setiap perkuliahan harus terdapat mahasiswa yang hadir minimal satu orang	[√] diterima [ ] ditolak

Aplikasi Pencetakan Laporan Berita Acara Perkuliahan

1. Pengujian Login

Tabel 4.9 Pengujian login aplikasi pencetakan laporan berita acara perkuliahan

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang**	**Pengamatan**	**Kesimpulan**

		diharapkan		
1.	NIP dan Password staf sekretariat jurusan, contoh : NIP : 99999999 Password : 1234	Masuk aplikasi pencetakan laporan Berita Acara Perkuliahan	Dapat masuk ke aplikasi pencetakan laporan Berita Acara Perkuliahan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dan Password, contoh : NIP : 2 Password : 1234	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
2.	NIP dan Password, contoh : NIP : 99999999 Password : 2	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
3.	NIP dan Password, contoh : NIP : 2 Password : 2	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak

2. Pengujian Menampilkan Laporan

Tabel 4.10 Pengujian Menampilkan Laporan

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dosen. Tahun dan semester akademik, contoh : NIP :	Memunculkan laporan Berita Acara Perkuliahan dosen dengan NIP 41277006009,	Muncul laporan Berita Acara Perkuliahan dosen tersebut	[√] diterima [ ] ditolak

	41277006009 Tahun akademik : 2011/2012 Semester : genap	tahun akademik 2011/2012 semester genap		
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dosen. Tahun dan semester akademik, contoh : NIP : 1 Tahun akademik : 2011/2012 Semester : genap	Muncul pesan "data dosen tidak ditemukan"	Muncul pesan "data dosen tidak ditemukan"	[√] diterima [ ] ditolak
2.	NIP dosen. Tahun dan semester akademik, contoh : NIP : 1 Tahun akademik : 2011/2012 Semester : ganjil	Muncul pesan "data BAP tidak ditemukan"	Muncul pesan "data BAP tidak ditemukan"	[√] diterima [ ] ditolak

Aplikasi Kuliah Pengganti

1. Pengujian Login

Tabel 4.11 Pengujian login aplikasi kuliah pengganti

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dan Password dosen, contoh : NIP : 41277006009	Masuk aplikasi pengisian Berita Acara Perkuliahan	Dapat masuk ke aplikasi pengisian Berita Acara Perkuliahan	[√] diterima [ ] ditolak

	Password : irawan			
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	NIP dan Password, contoh : NIP : 1 Password : irawan	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
2.	NIP dan Password, contoh : NIP : 41277006009 Password : taryana	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak
3.	NIP dan Password, contoh : NIP : 1 Password : 1	Muncul pesan kesalahan "NIP atau Password salah!"	Muncul pesan kesalahan	[√] diterima [ ] ditolak

2. Pengujian Mengisi Data Kuliah Pengganti

Tabel 4.12 Pengujian mengisi data kuliah pengganti

Kasus dan hasil uji (data benar)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tanggal pelaksanaan (tanggal belum lewat dan bukan hari minggu), ruang yang akan digunakan, jam yang menyediakan SKS yang	Muncul pesan "SUKSES : Data kuliah pengganti berhasil disimpan"	Muncul pesan sukses	[√] diterima [ ] ditolak

	cukup, matakuliah, dan kelas yang akan diajar, contoh : Tanggal : 3-Agustus-2012 Ruang : LAB-5 Jam : 19.45 (menyediakan 4 SKS) Matakuliah : IF38346P – Database Lanjut (3 SKS) Kelas : IF-10 Semester VIII			
Kasus dan hasil uji (data salah)				
No	**Data Masukan**	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1.	Tanggal pelaksanaan (sudah lewat), contoh : Tanggal : 27-Juli-2012	Muncul peringatan "tidak bisa memilih tanggal yang sudah lewat"	Muncul peringatan "tidak bisa memilih tanggal yang sudah lewat"	[√] diterima [ ] ditolak
2.	Tanggal pelaksanaan (hari minggu), contoh : Tanggal : 5-Agustus-2012	Muncul peringatan "tidak bisa memilih hari minggu"	Muncul peringatan "tidak bisa memilih hari minggu"	[√] diterima [ ] ditolak
3.	Jam tidak menyediakan SKS yang cukup, contoh : Tanggal : 3-agustus-2012 Ruang : LAB-5 Jam : 21.15 (menyediakan 2 SKS)	Muncul peringatan "Waktu yang dipilih tidak mencukupi untuk melakukan perkuliahan"	dipilih tidak mencukupi untuk melakukan perkuliahan"	[√] diterima [ ] ditolak

	Matakuliah : IF38346P – Database Lanjut (3 SKS)			
4.	Data kuliah pengganti yang dimasukkan bentrok dengan jadwal kuliah mahasiswa, contoh : Tanggal : 4-Agustus-2012 Ruang : LAB-5 Jam : 13.00 (menyediakan 13 SKS) Matakuliah : IF38346P – Database Lanjut (3 SKS) Kelas : IF-10 Semester VIII	Muncul peringatan "Jadwal Bentrok"	Muncul peringatan "Jadwal Bentrok"	[√] diterima [ ] ditolak
5.	Tanggal : 3-Agustus-2012 Ruang : **(kosong)** Jam : 19.45 (menyediakan 4 SKS) Matakuliah : IF38346P – Database Lanjut (3 SKS) Kelas : IF-10 Semester VIII	Muncul peringatan "Ruangan belum dipilih"	Muncul peringatan "Ruangan belum dipilih"	[√] diterima [ ] ditolak
6.	Tanggal : 3-Agustus-2012 Ruang : LAB-5 Jam : **(kosong)** Matakuliah :	Muncul peringatan "jam belum dipilih"	Muncul peringatan "jam belum dipilih"	[√] diterima [ ] ditolak

	IF38346P – Database Lanjut (3 SKS) Kelas : IF-10 Semester VIII			
7.	Tanggal : 3-Agustus-2012 Ruang : LAB-5 Jam : 19.45 (menyediakan 4 SKS) Matakuliah : IF38346P – Database Lanjut (3 SKS) Kelas : **(kosong)**	Muncul peringatan "Kelas belum dipilih"	Muncul peringatan "Kelas belum dipilih"	[√] diterima [ ] ditolak

4.1.2.3 Kesimpulan pengujian Alpha

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa sistem monitoring penjadwalan ruang kelas dan Berita Acara Perkuliahan telah bebas dari kesalahan penulisan, dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan .

4.3.1 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana aplikasi digunakan secara langsung oleh pengguna yaitu dosen dan staf sekretariat jurusan, untuk mengetahui kepuasan user dan kandungan *point* syarat *user friendly*. Pengujian betha dilakukan dengan metode wawancara, dengan pertanyaan sebagai berikut :

1. Wawancara Dosen

1. Apakah aplikasi ini membantu meningkatkan pengawasan terhadap penggunaan ruang kelas dan pengisian Berita Acara Perkuliahan ?
2. Bagaimana menurut anda instruksi yang ada pada aplikasi ini, mudah atau sulit dimengerti ?
3. Apakah bapak/ibu dosen setuju apabila aplikasi ini diterapkan di Universitas ?
4. Apakah aplikasi ini sudah layak untuk digunakan ?

2. Hasil Wawancara Dosen

Berikut adalah hasil wawancara dengan dosen UNIKOM :

Nama : Irawan Afrianto, S.T., M.T.

NIP : 41277006009

Jurusan: Teknik Informatika

1. Apakah aplikasi ini membantu meningkatkan pengawasan terhadap penggunaan ruang kelas dan pengisian Berita Acara Perkuliahan ?

   Jawab : "Ya, dengan adanya prototipe aplikasi ini, proses monitoring terhadap ruangan menjadi lebih teratur dan memudahkan dosen dalam pengisian BAP."

2. Bagaimana menurut anda instruksi yang ada pada aplikasi ini, mudah atau sulit dimengerti ?

Jawab : "Mudah, tampilan menu sudah mempermudah dosen untuk memahami sistem. Hanya saja perlu ditambahkan bantuan/help u/ lebih memberikan penjelasan tentang aplikasi."

3. Apakah bapak/ibu dosen setuju apabila aplikasi ini diterapkan di Universitas ?

Jawab : "Setuju, sebagai intstitusi pendidikan tinggi berbasis ICT, sudah saatnya menerapkan teknologi yang dapat memberikan kemudahan/efisien dan efektif dalam KBM."

4. Apakah aplikasi ini sudah layak untuk digunakan ?

Jawab : "Untuk sebagai prototipe sudah sangat baik, u/ implementasi perlu dikaitkan dengan p/k sebenarnya."

Nama : Robi Tanzil G, S.Kom.

NIP : -

Jurusan: Teknik Informatika

1. Apakah aplikasi ini membantu meningkatkan pengawasan terhadap penggunaan ruang kelas dan pengisian Berita Acara Perkuliahan ?

Jawab : "Ya, membantu

2. Bagaimana menurut anda instruksi yang ada pada aplikasi ini, mudah atau sulit dimengerti ?

Jawab : "Cukup mudah digunakan."

3. Apakah bapak/ibu dosen setuju apabila aplikasi ini diterapkan di Universitas ?

Jawab : "Ya, sangat setuju."

4. Apakah aplikasi ini sudah layak untuk digunakan ?

Jawab : "Layak..."

3. Wawancara Staf Sekretariat Jurusan

1. Apakah aplikasi ini membuat pencetakan laporan BAP menjadi lebih mudah?
2. Menurut Anda, apakah aplikasi ini mudah digunakan?
3. Apakah bapak/ ibu staf sekretariat jurusan setuju apabila aplikasi ini diterapkan di universitas?
4. Apakah aplikasi ini sudah layak untuk digunakan?

4. Hasil Wawancara Staf Sekretariat Jurusan

Berikut adalah hasil wawancara dengan staf sekretariat jurusan UNIKOM.

Nama : Hendar Rubedo

NIP : 41271001093

Jurusan: Teknik Informatika

1. Apakah aplikasi ini membuat pencetakan laporan BAP menjadi lebih mudah?

Jawab : "Sebenarnya pencetakan BAP pada sistem yang sedang berjalan ini tidak sulit. Yang sulit adalah mengumpulkan BAP para dosen tiap minggunya. Jadi, jika dibandingkan dengan pencetakan BAP-nya saja, mekanisme pencetakan BAP yang diusulkan tidak/ masih belum membuat pencetakan BAP menjadi lebih mudah."

2. Menurut Anda, apakah aplikasi ini mudah digunakan?

Jawab : "cukup mudah dimengerti dan digunakan. Hanya saja akan merepotkan jika mencetaknya satu persatu dan berdasarkan NIP dosen. Artinya sekretariat jurusan harus mengetahui NIP masing-masing dosen yang akan dicetak BAP-nya"

3. Apakah bapak/ ibu staf sekretariat jurusan setuju apabila aplikasi ini diterapkan di universitas?

Jawab : "Setuju. Namun perlu pembenahan lebih lanjut agar bisa diterima semua komponen/ personal yang terlibat"

4. Apakah aplikasi ini sudah layak untuk digunakan?

Jawab : "Belum. Mekanisme pencetakan BAP perlu dibenahi lagi dan disesuaikan dengan mekanisme pengisian BAP atau aturan perkuliahan yang baru apabila sistem ini diterapkan.

4.3.2 Kesimpulan pengujian Beta

Dari hasil pengujian betha yang telah dilakukan, dapat ditarik kesimpulan bahwa :

1. Aplikasi ini membuat penggunaan ruang kelas lebih teratur sesuai jadwal yang sudah dibuat.
2. Aplikasi ini memudahkan dosen dalam melakukan pengisian Berita Acara Perkuliahan.
3. Aplikasi ini membuat data BAP yang dihasilkan menjadi lebih akurat, namun belum memudahkan staf sekreteriat jurusan dalam mencetak laporan BAP

yang diperlukan. Hal ini disebabkan pencetakan laporan BAP dilakukan satu-persatu dan harus memasukkan NIP dosen yang akan dicetak BAP-nya. cara ini dianggap melelahkan jika akan mencetak BAP dosen dalam jumlah banyak dan dilakukan tiap minggu.