



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

3.1. Gambaran Umum Perusahaan

Merupakan sebuah lembaga pendidikan yang didirikan pada 25 November 2005 berdasarkan Surat Keputusan Menteri Pendidikan Nasional No. 169/D/O/2005 yang operasionalnya secara resmi dikelola oleh **Yayasan Multimedia Nusantara**. Yayasan ini didirikan oleh **Kompas Gramedia**, suatu kelompok usaha terkenal yang bergerak di bidang media massa, penerbitan, percetakan, toko buku, hotel, dan jasa pendidikan.

UMN merupakan sebuah lembaga perguruan tinggi dengan teknologi informasi dan komunikasi sebagai dasar daripada setiap proses belajar mengajar pada setiap mata kuliah yang diselenggarakannya. Didukung oleh keberadaan para tenaga pengajar yang profesional dan berpengalaman dalam bidang pendidikan serta penyelenggaraan program mata kuliah yang terarah dan terintegrasi akan mengantarkan **UMN** menjadi universitas unggulan di tingkat nasional maupun internasional. **UMN** disasarkan menjadi inspirasi bagi hadirnya paradigma pendidikan baru bagi kaum muda Indonesia sehingga mampu menghasilkan lulusan berkompetensi tinggi dan berjiwa wirausaha berbasis teknologi (*technopreneurship*).

UMN didirikan dengan ide dari Dr. (HC) Jakob Oetama sebagai perintis dari Kompas-Gramedia Group dan ditangani oleh Prof. Yohanes Surya, Ph.D,

sebagai rektor pertama dari **UMN**. Pada tanggal 25 November 2005, ijin operasional dari **UMN** dikeluarkan oleh menteri pendidikan nasional RI. Lalu pada 20 November 2006, **UMN** secara resmi diluncurkan dalam sebuah acara pada **Hotel Santika**. Pada 14 Januari 2007, ujian saringan masuk **UMN** yang pertama diselenggarakan di dalam gedung Kompas Gramedia Unit III. Tanggal 9 Juli 2007, program matrikulasi pertama untuk angkatan pertama **UMN** diselenggarakan pada kampus pertama di **Wisma 46**, Jakarta. Tanggal 31 Agustus hingga 1 September 2007 diselenggarakan Orientasi Mahasiswa Baru yang pertama dalam **Wisma Kompas Pacet**, Jawa Barat. Kemudian, kuliah perdana bagi angkatan pertama dengan tema “Pengembangan Sumber Daya Manusia Menyongsong Era ICT” diselenggarakan pada 3 September 2007. Pada 5 September 2008 dimulailah pembangunan kompleks kampus **UMN** di daerah Gading Serpong, Tangerang. Mulai pada tanggal 24 Agustus 2009, **UMN** pindah ke kompleks kampus di Gading Serpong, Tangerang. Pada tanggal 26 Agustus 2009, OMB tahun ajaran 2009/2010 dibuka dalam kampus **UMN** dengan pemimpin acara rektor **UMN**, Prof. Yohanes Surya, Ph.D. Tanggal 2 Desember 2009, Kampus **UMN** yang terletak di daerah Scientia Garden, Summarecon Serpong diresmikan. Pada 25 Maret 2011, Dr. Ninok Leksono naik menjadi rektor **UMN** menggantikan posisi Prof. Yohanes Surya, Ph.D. Kampus **UMN** tahap 2 mulai dibangun pada 29 April 2011. Pada 26 November 2011, wisuda pertama **UMN** yang diikuti oleh 75 orang wisudawan dilaksanakan. Tanggal 8 Februari 2012, kampus **UMN** tahap 2 dibuka dan mulai dipakai untuk acara perkuliahan.

UMN sendiri memiliki visi untuk menjadi sebuah perguruan tinggi yang unggul dalam bidang ICT baik itu dalam tingkat nasional maupun internasional dan juga dapat menghasilkan lulusan berwawasan internasional dan berkompentensi tinggi pada bidangnya (berkeahlian tinggi) dengan jiwa wirausaha serta berbudi pekerti luhur. Selain itu, UMN memiliki misi untuk ikut serta dalam proses mencerdaskan kehidupan bangsa dan memajukan kesejahteraan bangsa melalui upaya penyelenggaraan pendidikan tinggi dengan melaksanakan Tridarma Perguruan Tinggi (Pendidikan, Penelitian dan Pengabdian pada Masyarakat), untuk meningkatkan kualitas sumber daya manusia Indonesia.

Beberapa keunggulan dari UMN itu sendiri adalah dalam :

1. **Fokus dalam bidang teknologi informasi dan komunikasi (TIK)**
2. **Komitmen dari para pendiri** untuk menjadikan UMN menjadi sebuah universitas unggulan
3. **Technopreneurship**, lulusan dari UMN diharapkan tidak hanya siap untuk menjadi sebuah profesional dan hanya siap untuk bekerja kepada orang lain melainkan juga dapat menciptakan sebuah lapangan kerja baru bagi para calon pekerja lainnya
4. **Lokasi dan Fasilitas**, dengan kampus yang terletak di dalam sebuah pemukiman elit dan strategis di wilayah **Gading Serpong**, diharapkan UMN juga dapat ikut berkembang secepat perkembangan dari wilayah sekitarnya
5. **Kualitas dari para lulusannya**, setiap mahasiswa UMN dibimbing agar tidak hanya dapat bekerja pada orang lain. Mereka

juga dibimbing agar dapat bekerja sendiri dan membuat lapangan kerja baru bagi para orang lainnya. Para mahasiswa **UMN** sendiri dibimbing agar menjadi seseorang yang memiliki kualitas tinggi agar dapat bersaing tidak hanya untuk bekerja di dalam negeri, tetapi juga untuk bekerja diluar negeri.

Untuk **BIPA UMN**, yang merupakan singkatan dari **Bahasa Indonesia untuk Penutur Asing UMN**, sendiri merupakan sebuah bagian dari **UMN** itu sendiri. Disini, **BIPA UMN** memberikan pelajaran mengenai Bahasa Indonesia yang diberikan kepada berbagai calon penutur asing dari luar negeri. Untuk saat ini, kebanyakan pelajar dari **BIPA UMN** merupakan pelajar dari korea.

BIPA UMN merupakan sebuah bagian dari *Continues Learning Department* yang ada di dalam **Universitas Multimedia Nusantara** itu sendiri. **CLD UMN** terbagi atas 3 bagian, yaitu : **BIPA UMN**, Bahasa Korea, dan Persiapan **TOEIC**. **BIPA UMN** menangani para orang asing yang tertarik akan **Bahasa Indonesia** dan berminat untuk mendalaminya. **CLD** dipimpin oleh seorang **CLD Manager**, yaitu **Bpk. Arief Setyadi**. Di **BIPA UMN** sendiri dapat dilihat, bahwa ada cukup banyak pengajar disana setiap harinya. Bahkan, pada saat kelas sedang dalam proses belajar-mengajar, ruang **BIPA UMN** tidak pernah kosong, selalu ada orang disana, baik itu seorang pengajar, ataupun orang yang sedang melakukan proses magang atau tugas akhir.

BIPA terdiri atas 3 tingkatan dan 2 program pengajaran. Tingkatan pengajarannya adalah Pemula, Madya, dan Mahir. Sedangkan untuk program pengajarannya terdiri atas reguler dan percakapan.

3.2. Struktur Organisasi

UMN sendiri tersusun antara beberapa tingkatan dengan yang tertinggi Rektor yang ditempati oleh **Dr. Ninok Leksono**, kemudian diisi dengan 4 orang Wakil Rektor, ada untuk **Bidang Akademik, Bidang Kemahasiswaan, Bidang Administrasi Umum dan Keuangan**, dan **Bidang Hubungan dan Kerjasama**. Lalu dibawahnya ada 4 orang Dekan dari 4 Fakultas berbeda, yaitu dari Fakultas **TIK, Komunikasi, Ekonomi, dan Seni & Desain**. Kemudian dibawah para Dekan tersebut ada para Kepala Program Studi untuk setiap program studi. Dalam UMN sendiri, saat ini ada 7 program studi dari 4 fakultas yang berbeda, yaitu :

1. Fakultas **Teknik Informasi dan Komunikasi**
 - a. Program Studi **Teknik Informatika**.
 - b. Program Studi **Sistim Informasi**.
 - c. Program Studi **Sistem Komputer**.
2. Fakultas **Seni dan Desain**
 - a. Program Studi **Desain Komunikasi dan Visual**.
3. Fakultas **Komunikasi**
 - a. Program Studi **Ilmu Komunikasi**.
4. Fakultas **Komunikasi**
 - a. Program Studi **Manajemen**.
 - b. Program Studi **Akuntansi**.

Lalu, untuk **BIPA UMN**, yang merupakan sebuah organisasi tambahan di dalam UMN itu sendiri, adalah sebuah organisasi yang ada untuk menyalurkan Bahasa Indonesia kepada berbagai pihak, khususnya para calon-calon penutur

asing dan juga orang-orang yang ingin mendalami Bahasa Indonesia itu sendiri.

BIPA UMN berdiri pada tahun 2012 dan saat ini dipimpin oleh Ibu Tria.

3.3. *Business Process* Absensi Pelajar Pada BIPA UMN Yang Sedang Berjalan Saat Ini

Di **BIPA UMN** sendiri yang untuk melakukan absensi terhadap para pelajarnya saat ini masih menggunakan proses manual. Bahkan para data pelajarnya juga masih menggunakan proses manual. Proses manual yang dimaksud adalah absensi menggunakan kertas dan tanda tangan dari para pelajarnya pada saat melakukan proses absensi

Proses ini dimulai sejak **BIPA UMN** mulai berjalan. Dengan sistem yang berjalan saat ini, **BIPA UMN** akan membutuhkan banyak kertas yang akan terpakai untuk mencatat data para pelajar saat mereka mendaftarkan diri mereka sebagai pelajar pada **BIPA UMN**.

Dengan sistem yang ada saat ini, kebutuhan akan kertas dapat terus meningkat seiring dengan semakin bertambahnya jumlah pelajar **BIPA UMN**. Selain itu, dengan sistem yang berlangsung saat ini, data kehadiran dan data lengkap para pelajar dapat saja dilihat dan juga diubah oleh pihak yang tidak seharusnya melakukan hal tersebut. Data-data tersebut juga dapat saja disebarluaskan atau dijual kepada pihak-pihak lainnya. Sehingga keamanan dari data milik **BIPA UMN** dapat dikategorikan mudah untuk dirusak / *vulnerable*.

3.4. *Business Process* Yang Diusulkan Oleh Penulis

Pada *business process* atau sistem yang baru, penulis mengusulkan agar pihak pengelola **BIPA UMN** untuk mengganti sistem absensi yang dipakai saat ini dengan sebuah kartu yang di dalamnya sudah ditanamkan sebuah *chip* **RFID**.

Penulis mengusulkan menggunakan sistem **RFID** dibandingkan dengan sistem lainnya, seperti barcode dan sidik jari karena sistem **RFID** tidak memakan biaya yang besar seperti sistem sidik jari dan tidak mudah diduplikasi seperti sistem barcode. Lalu, bila pada sistem terdahulu setiap pelajar akan membutuhkan waktu agak banyak untuk mencari namanya, mengeluarkan alat tulis, dan menandatangani lembar absensi tersebut, sehingga bila proses absensi yang lama dilakukan pada waktu kelas berlangsung, proses absensi sendiri akan memakan waktu belajar-mengajar. Tetapi, dengan menggunakan sistem **RFID**, proses absensi dapat dilakukan dengan lebih mudah dan cepat. Para pelajar hanya perlu untuk melakukan sekali proses *tapping* dengan mendekatkan kartu **RFID** milik mereka dengan alat pembaca kartu **RFID** kita. Selain itu, pada saat ini, data lengkap para pelajar dapat diakses untuk diubah maupun hanya melihatnya oleh siapa saja yang mengetahui lokasi penyimpanan data tersebut. Bahkan, pada sistem yang ada saat ini, data dapat diduplikasi oleh siapa saja. Tetapi, dengan sistem yang baru, data hanya dapat dilihat, ditambah, dan diubah oleh pihak yang memang memiliki kewenangan untuk melakukannya.

Lalu, bila pada sistem yang lama, setiap orang dapat saja meminta temannya untuk melakukan absensi dirinya. Pada sistem yang baru, kejadian seperti meminta agar absensi dilakukan oleh orang lain akan menjadi lebih susah

untuk dilakukan. Hal ini dikarenakan mereka harus memiliki kartu **RFID** milik pelajar yang ingin agar dirinya diabsensi oleh orang lain. Sebuah hal yang dapat menjadi lebih sulit karena kartu **RFID** bukan sebuah kartu yang dapat didapatkan begitu saja. Lalu, dengan data unik yang umumnya sudah tercatat pada setiap kartu **RFID**, maka data dari kartu mereka harus didaftarkan dahulu kedalam *database* pelajar **BIPA UMN**.

Berbagai data mengenai para pelajar **BIPA UMN** dan juga data mengenai kehadiran mereka serta data para pengguna sistem umumnya hanya dapat diakses oleh beberapa pihak yang telah didaftarkan sebelumnya. Saat ini, hanya para administrator sistem yang dapat mengakses seluruh data mengenai **BIPA UMN** itu sendiri. Para pengguna biasa hanya dapat melihat data kehadiran pelajarnya.

Apabila ada kasus dimana kartu milik pelajar hilang ataupun rusak, pelajar tersebut dapat segera melaporkan ke pihak **BIPA UMN** untuk diurus. Nantinya, data di dalam *database* pelajar **BIPA UMN** akan di-*update* sehingga data yang tersimpan nantinya akan disesuaikan dengan kartu barunya.

Lalu, untuk merancang sistem ini, penulis memilih menggunakan bahasa pemrograman **C++** karena mendukung teknologi **RFID** selain itu dibandingkan dengan berbagai bahasa pemrograman lainnya, seperti *Visual Basic* dan *Java*, hasil kompilasi dari bahasa **C++** sendiri jauh lebih kecil daripada bila penulis memakai bahasa pemrograman *Visual Basic* dan *Java*. Dan dengan hasil kompilasi yang kecil, maka sistem tidak akan memakan memori dari sistem operasi yang besar dan dapat berjalan dalam waktu yang singkat. Penulis sendiri pernah memakai ketiga bahasa pemrograman tersebut dan mendapatkan bahwa

bila untuk sebuah aplikasi kecil, seperti kalkulator simpel, pada bahasa **C++** outputnya berukuran 160 kb, lalu pada bahasa *Visual Basic* berukuran 349 kb dan pada *Java* berukuran 250 kb.

3.5. Metode Perancangan Sistem

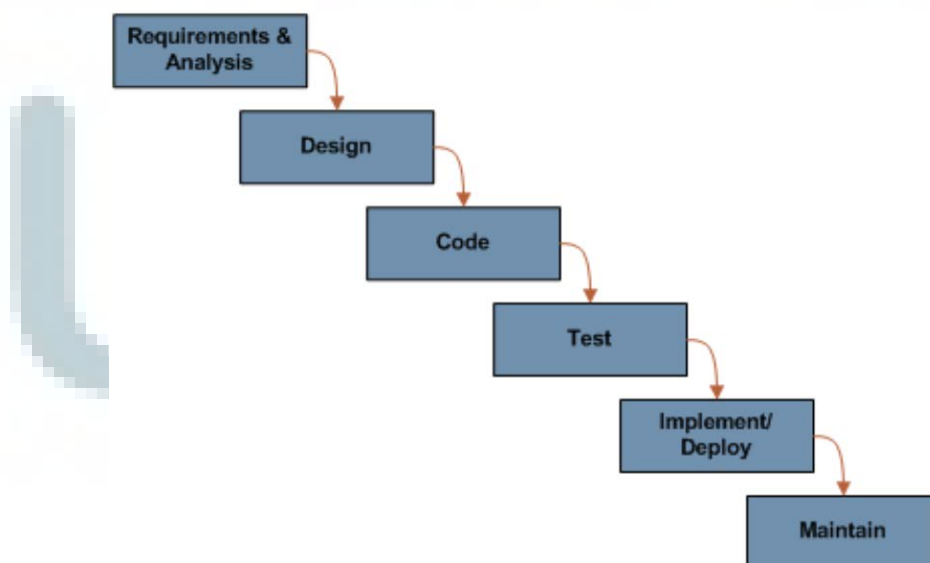
Dalam menerapkan metode baru ini, perancangan sistem yang akan dipakai dapat menggunakan berbagai macam metode. Penulis sendiri memilih untuk memakai metode *waterfall* (Air Terjun). Penulis memilih metode air terjun karena seluruh bagian dalam metode ini akan dijalankan secara sistematis melalui tahapan-tahapan yang telah tersusun sebelumnya tanpa ada yang terlewat. *Waterfall model* juga dipilih karena merupakan metode yang populer dan simpel sehingga proses pengembangan sistem dapat dilakukan dengan lebih mudah.

Waterfall model merupakan sebuah model yang sering dipakai dalam proses pengembangan sebuah sistem. *Waterfall model* juga memberikan gambaran mengenai proses pengembangan sistem secara berurutan dimana suatu proses tidak dapat dimulai sebelum proses sebelumnya selesai. Berikut adalah langkah-langkah di dalam *Waterfall model*.

1. **Requirement & Analysis** - Pengumpulan apa yang sebenarnya diinginkan oleh calon pengguna nantinya serta melakukan analisa terhadap proses bisnis yang sedang berjalan.
2. **Design** - Membuat sebuah rancangan baru untuk memberikan masukan baru kepada pihak organisasi.

3. **Code** - Membuat aplikasi yang nantinya akan dipakai dalam rancangan baru tersebut dengan bantuan beberapa pihak lain dalam hal *coding*.
4. **Test** - Melakukan *testing* kepada sistem yang baru dibangun tersebut.
5. **Implement** - Proses yang baru akan diterapkan di dalam **BIPA UMN** dan para pengguna sistem serta para pelajar akan diajarkan bagaimana cara untuk menggunakan sistem absensi yang baru.
6. **Maintain** - Pemeliharaan terhadap sistem yang baru tersebut akan dimulai untuk menjaga agar kondisi sistem selalu dalam kondisi yang nyaris sempurna. Tidak hanya itu, dalam tahap ini, apabila ditemukan adanya kesalahan dan perlu melakukan *update*, maka *update* akan ditambahkan secepat mungkin.

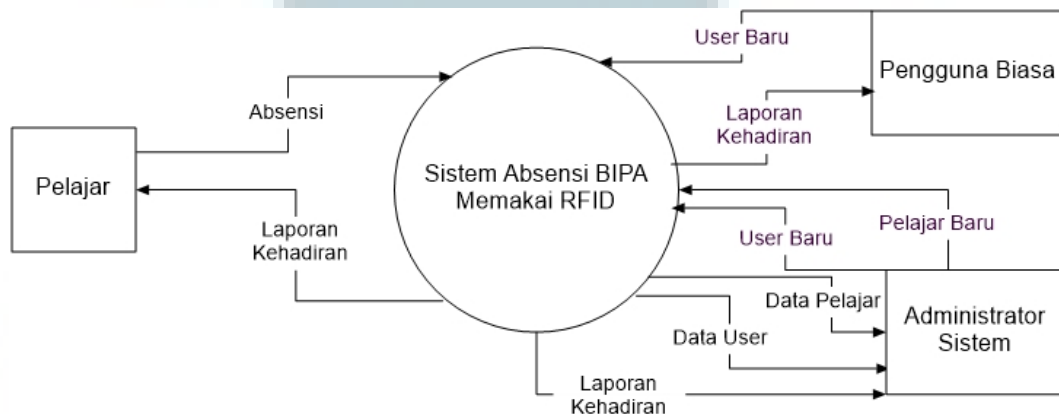
Metode ini sendiri akan dilakukan sesuai dengan yang tergambar dibawah.



Gambar 3.1. Model SDLC Bentuk Waterfall

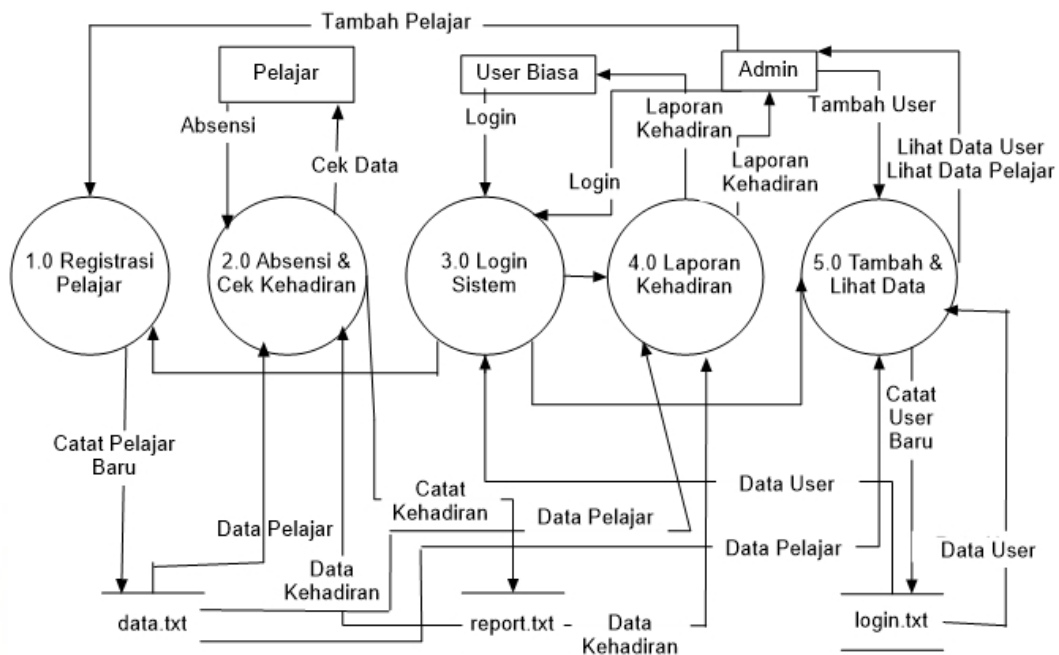
3.6. Diagram Arus Data / *Data Flow Diagram* (DAD / DFD)

Semua diagram arus data diawali sebuah diagram kompleks yang menghubungkan DAD yang ada menjadi satu. Berikut diagram kompleksnya.



Gambar 3.2. Diagram Kompleks Sistem

1. Administrator sistem mendaftarkan pelajar-pelajar baru **BIPA UMN** melalui sistem.
2. Para pelajar yang sudah didaftarkan dapat melakukan proses absensi dan melihat data kehadiran mereka memakai kartu **RFID**-nya.
3. Para administrator sistem dan juga para pengguna biasa dapat melihat laporan kehadiran para pelajar **BIPA UMN** baik itu laporan secara lengkap, laporan untuk hari tersebut, maupun laporan dari setiap kelasnya.
4. Para administrator sistem dan juga para pengguna biasa dapat mendaftarkan pengguna yang baru walaupun pengguna biasa hanya dapat mendaftarkan pengguna biasa lainnya sedangkan para administrator sistem dapat juga mendaftarkan administrator sistem yang baru.
5. Para administrator sistem dapat melihat data lengkap para pelajar **BIPA UMN** dan data lengkap para pengguna sistem.



Gambar 3.3. DAD Level 0

Proses 1.0 – Semuanya dimulai dengan pendaftaran para pelajar oleh pihak administrator sistem. Setelah admin sistem memasukan data, sistem akan mencatat data tersebut pada data.txt.

Proses 2.0 – Para pelajar yang sudah ada datanya pada data.txt dapat melakukan proses absensi dengan memakai kartu RFID mereka. Pada proses absensi, sistem akan menambahkan data mereka setelah sistem memastikan bahwa mereka belum melakukan absensi pada hari tersebut. Data akan dicatat oleh sistem pada report.txt. Mereka juga dapat melihat data kehadiran mereka yang tercatat sebelumnya.

Proses 3.0 – Para user biasa dan administrator yang ingin menggunakan sistem diwajibkan untuk melakukan login terlebih dahulu. Sistem akan mengecek data yang mereka masukan di halaman login dengan data yang tersimpan di login.txt. Bila data tidak sama, sistem akan kembali ke halaman utama.

Proses 4.0 – Para user biasa dan administrator dapat melihat data kehadiran para pelajar dengan menggabungkan data yang ada di data.txt dan pada report.txt. Laporan yang ditampilkan dapat berupa laporan lengkap kehadiran dari awal, laporan kehadiran pada hari tersebut, maupun laporan kehadiran dari tiap kelas.

Proses 5.0 – Para admin sistem dapat menambah data para pengguna sistem bertipe admin maupun user biasa yang akan dicatat sistem pada login.txt lalu para admin sistem juga dapat melihat data siapa saja pelajar **BIPA UMN** dan siapa saja pengguna user hingga saat itu.

3.7. Entity Relationship Diagram (ERD)

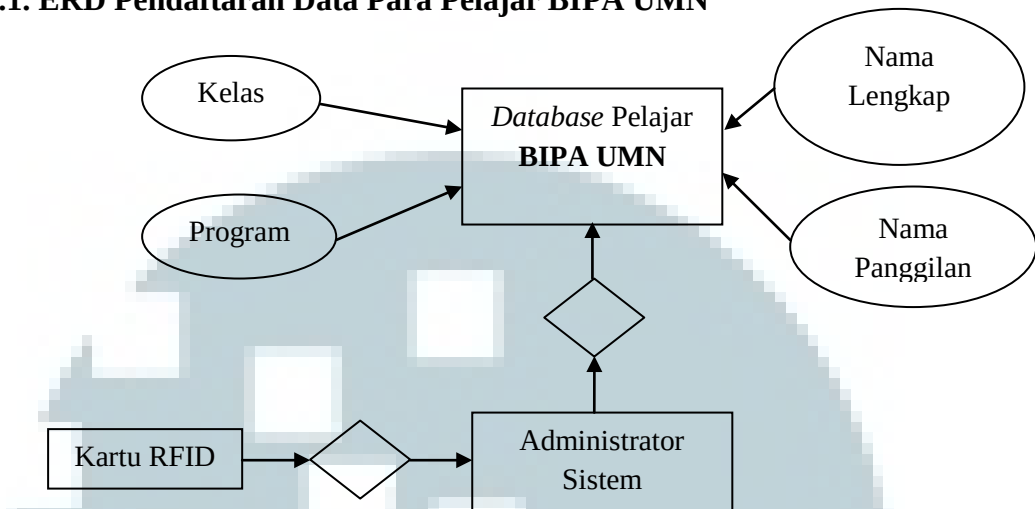
Disini penulis akan menampilkan bagaimana sistem berjalan, bagaimana sistem yang dibuat penulis berlangsung, serta apa saja yang terdapat di dalam sistem yang dibuat oleh penulis itu sendiri. Dalam setiap **ERD** yang dibuat oleh penulis itu sendiri juga akan ada penjelasan mengenai apa maksud dari bagian ini, mengapa ada, dan untuk apa ini ada.

Dalam bagian ini, akan ada 4 buah **ERD**, yaitu :

1. **ERD** untuk pendaftaran data para pelajar dari **BIPA UMN**.
2. **ERD** mengenai bagaimana seseorang akan dapat melakukan proses absensi.
3. **ERD** mengenai bagaimana proses dimana seorang pengguna aplikasi yang menginginkan data kehadiran dari seluruh pelajar.
4. **ERD** mengenai bagaimana proses dimana seorang pelajar menginginkan data kehadiran dirinya.

Kita akan mulai secara berurutan mulai dari yang pertama.

3.7.1. ERD Pendaftaran Data Para Pelajar BIPA UMN

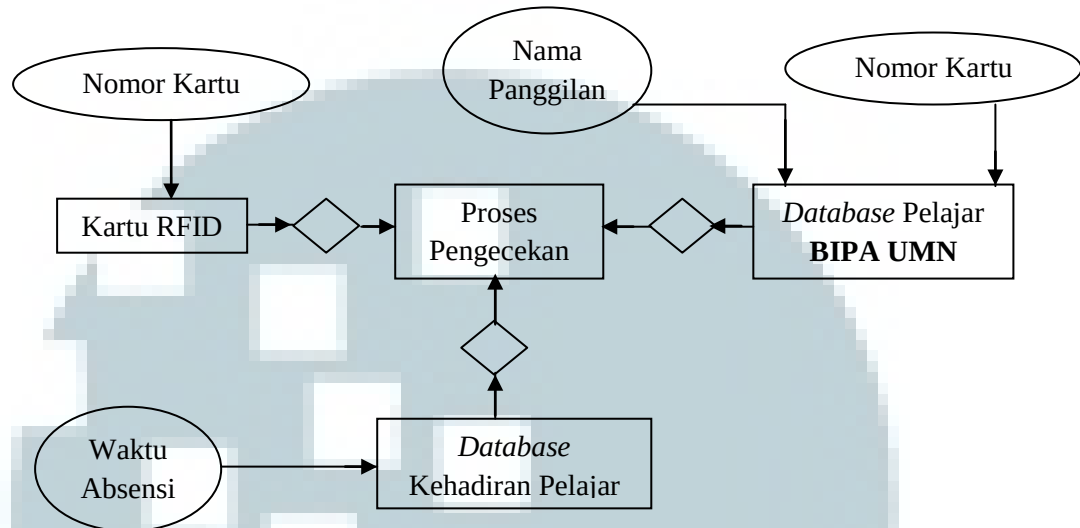


Gambar 3.4. ERD Pendaftaran Data Pelajar BIPA UMN

Disini, setelah kartu **RFID** pelajar dibaca oleh alat pembaca chip **RFID**, dan data mahasiswa secara lengkap didapatkan, data tersebut akan dikirimkan kepada Administrator Sistem untuk kemudian dimasukkan ke dalam *database* Pelajar **BIPA UMN**. Di dalam *database* tersebut tersimpan nomor kartu **RFID** yang mereka dapatkan, nama lengkap / nama asli mereka, nama Indonesia / panggilan, kelas yang dijalani, dan program mereka saat ini.

Disini terdapat variabel nama panggilan yang akan sering dibaca pada saat-saat tertentu, seperti pada waktu sistem memasukkan data pada *database* kehadiran pelajar sehingga pada waktu laporan diminta, entah oleh Administrator Sistem ataupun oleh pihak pengelola sistem datang, maupun saat diminta oleh pelajarnya sekalipun, pihak yang meminta laporan tersebut tidak akan bingung siapa orang itu.

3.7.2. ERD Bagaimana Seseorang Dapat Melakukan Proses Absensi



Gambar 3.5. ERD Proses Absensi Pelajar

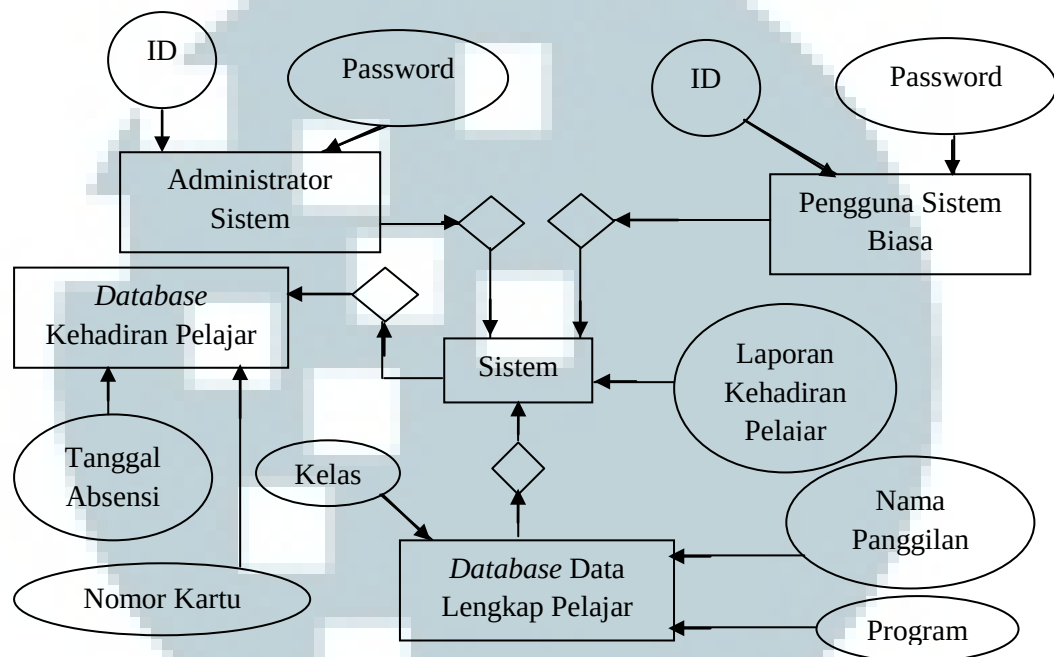
Untuk bagian yang satu ini, pada waktu pelajar melakukan *tapping* kartu **RFID** yang ia miliki ke alat pembaca chip **RFID**, alat tersebut akan mengirimkan data kartu tersebut untuk dibandingkan dengan *database* yang menyimpan data lengkap pelajar **BIPA UMN**. Pada proses pengecekan, sistem akan membandingkan nomor kartu milik pelajar yang meminta agar ia dianggap hadir dengan nomor kartu pada *database* yang mencatat data lengkap setiap pelajar.

Bila nomor kartu pelajar yang melakukan proses *tapping* sama dengan yang ada di *database*, sistem akan mencatatkan data kehadiran pelajar tersebut di dalam *database* kehadiran pelajar.

3.7.3. ERD Bagaimana Seseorang Dapat Mendapatkan Laporan Data Kehadiran Para Pelajar Secara Lengkap

Kemudian, mengenai ERD permintaan laporan kehadiran dari setiap pelajar mereka. Disini, sistem akan dapat diakses baik itu seorang administrator

sistem, maupun seorang pengguna sistem biasa. Sistem akan menampilkan sebuah laporan lengkap mengenai data kehadiran seluruh pelajar yang tercatat di dalam *database* kehadiran pelajar.

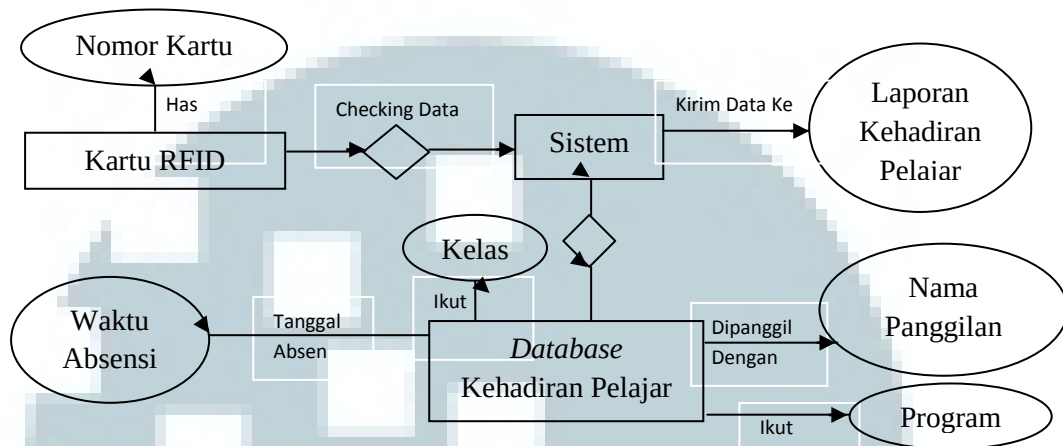


Gambar 3.6. ERD Laporan Lengkap Kehadiran Pelajar

Pertama-tama, orang yang ingin meminta laporan tersebut harus masuk ke dalam sistem lalu setelah masuk ke layar utama mereka, dimana bagi seorang administrator sistem akan masuk ke dalam layar utama administrator, sedangkan bagi pengguna sistem biasa akan masuk ke dalam layar utama user. Kemudian, mereka harus meminta agar dibukakan layar Laporan Kehadiran Para Pelajar yang akan menampilkan laporan mengenai siapa saja pelajar yang telah melakukan absensi, kapan ia melakukan absensi, juga pelajar tersebut ada di dalam kelas dan program apa.

3.7.4. ERD Bagaimana Seseorang Pelajar Dapat Mendapatkan Laporan Data

Kehadiran Dirinya



Gambar 3.7. ERD Laporan Kehadiran Seorang Pelajar

tertentu. Disini, laporan yang diberikan sistem hanya dapat diberikan pada pelajar yang memiliki kartu **RFID**. Sistem akan menampilkan sebuah laporan lengkap mengenai data kehadiran pelajar tersebut yang tercatat di dalam *database* kehadiran pelajar.

Pertama-tama, dari tampilan layar utama, sistem akan meminta agar pelajar tersebut melakukan proses *tapping* agar sistem mengetahui data yang mana yang harus ditampilkan. Lalu, dengan data kartu yang baru saja didapat, sistem akan membandingkan data tersebut dengan nomor kartu yang ada pada sistem. Setelah itu, sistem akan menampilkan data kehadiran dari pelajar yang melakukan proses *tapping* tadi.

3.8. Kamus Data

Di dalam sistem ini, terdapat 1 buah *database* bernama **BIPA** dengan 3 buah *file*, yaitu : *file* data.txt, *file* login.txt, dan *file* report.txt. Disini, penulis masih memakai sistem TXT untuk sistem *database*-nya.

Pada *file* data.txt, tercantum data mengenai nomor kartu **RFID** yang dipakai untuk absensi, nama asli atau nama lengkap mereka, nama panggilan mereka, jenis kelas yang mereka ambil, dan program apa yang sedang mereka ambil. Umumnya saat *file* diakses, data yang dilihat adalah data nomor kartu **RFID** yang mereka pakai karena tidak ada data tersebut yang bernilai sama. Data jenis kelas untuk saat ini hanya terisikan 4 jenis kelas, yaitu Pemula A, Pemula B, Madya, dan Mahir. Sedangkan data program hanya ada 2 program untuk saat ini, yaitu percakapan dan reguler.

Pelajar = No_Kartu + Nama_Lengkap + Panggilan + Jenis_Kelas + Program

Pada *file* login.txt, tercantum data mengenai id, password, dan jenis akun yang dimiliki. Data mengenai id pengguna merupakan data yang unik dan tidak sama. Sedangkan mengenai data password, walaupun unik masih tetap ada kemungkinan sama dengan milik id yang lain. Sedangkan, jenis akun hanya terdiri atas 2 pilihan, yaitu admin, yang dipakai oleh administrator sistem, dan user, yang dipakai oleh pengguna biasa.

Pengguna = ID + Jenis_Pemakai + Password

Kemudian, *file* report.txt, disini tercatat data mengenai nomor kartu, dan tanggal hadir. Lagi-lagi data nomor kartu merupakan data yang paling sering diakses karena merupakan data yang unik. Lalu, ada data mengenai tanggal

mereka melakukan proses absensi. Data ini dapat digunakan sebagai data untuk membantu sistem yang baru untuk menjadi sebuah sistem absensi elektronik yang baru dan bagus. Disini hanya disimpan data nomor kartu dan tanggal hadir untuk mempersulit apabila ada orang luar yang ingin mengubah laporan kehadiran.

$$\text{Laporan} = \text{No_Kartu} + \text{Tanggal_Hadir}$$

Terakhir, ada laporan lengkap dimana laporan lengkap merupakan gabungan dari ketiga *file* yang ada untuk menampilkan laporan kehadiran semua pelajar **BIPA UMN**. Laporan lengkap terdiri atas data singkat dan tanggal hadir mereka. Data singkat para pelajar terdiri atas nama lengkap, panggilan, jenis kelas, dan program yang sedang diikuti. Disini tidak disebutkan adanya nomor kartu karena nomor kartu juga tidak diketahui semua orang kecuali para administrator sistem.

$$\text{Laporan_Lengkap} = \text{Data_Singkat} + \text{Tanggal_Hadir}$$

3.9. Normalisasi Data

UNF

$$\text{Laporan_Lengkap} = \text{Data_Singkat} + \text{Tanggal_Hadir}$$

1NF

$$\text{Data_Singkat} = \text{Nama_Lengkap} + \text{Panggilan} + \text{Jenis_Kelas} + \text{Program}$$

$$\text{Pelajar} = \text{No_kartu} + \text{Nama_Lengkap} + \text{Panggilan} + \text{Jenis_Kelas} + \text{Program}$$

$$\text{Laporan_Lengkap} = \text{Panggilan} + \text{Jenis_Kelas} + \text{Program} + \text{Tanggal_Hadir}$$

2NF

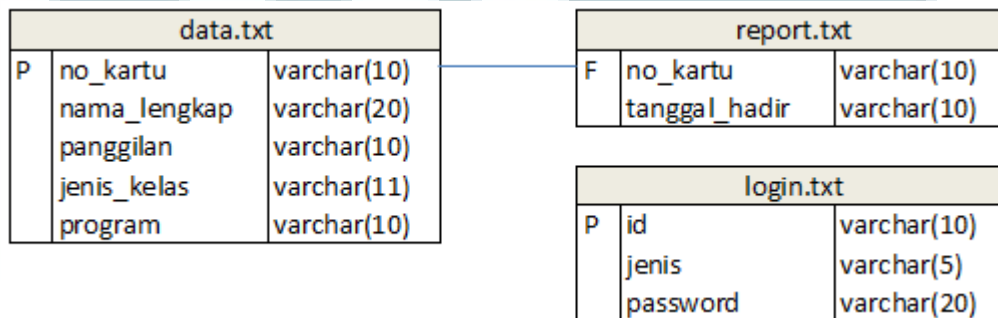
Jenis_Kelas = Tipe + No_Kelas

Data_Singkat = Panggilan + Tipe + No_Kelas + Program

Pelajar = No_kartu + Nama_Lengkap + Panggilan + Tipe + No_Kelas + Program

Laporan_Lengkap = Panggilan + Tipe + No_Kelas + Program + Tanggal_Hadir

3.10. Enhanced Entity Relationship (EER)



Gambar 3.8. EER Sistem BIPA UMN

Disini dijelaskan bahwa pada *database BIPA* nantinya akan terdapat 3 buah *file* yang akan dihubungkan dengan sistem yang baru. Di dalam sistem yang baru, data di dalam ketiga *file* tersebut dapat menjadi sebuah informasi dengan data yang tersimpan di tiga tempat berbeda. *File* yang paling sering diakses adalah *file data.txt* dan *report.txt* sedangkan *file login.txt* jarang karena sistem mengakses *login.txt* saat proses login berjalan dan saat proses tambah atau lihat data pengguna sistem berlangsung. Berikut adalah keterangan mengenai isi dari ketiga *file* tersebut.

File data.txt

Keterangan : berisikan data-data para pelajar **BIPA UMN**

Primary key : no_kartu

Kolom	Tipe Data
no_kartu	Varchar(10)
nama_lengkap	Varchar(20)
Panggilan	Varchar(10)
jenis_kelas	Varchar(11)
Program	Varchar(10)

File report.txt

Keterangan : berisikan data kehadiran pelajar **BIPA UMN**

Foreign key : no_kartu

Kolom	Tipe Data
no_kartu	Varchar(10)
tanggal_hadir	Varchar(10)

File login.txt

Keterangan : berisikan data-data para pengguna sistem

Primary key : id

Kolom	Tipe Data
Id	Varchar(10)
Jenis	Varchar(5)
password	Varchar(20)

3.11. Objek Implementasi

Objek yang dipakai untuk mengimplementasikan sistem ini adalah **BIPA Universitas Multimedia Nusantara** atau yang biasa disebut **BIPA UMN**. Implementasi sendiri akan dilakukan secara langsung oleh penulis dengan bantuan dari pihak **BIPA UMN** dan apabila penulis menemukan adanya masalah di dalam

sistem saat dilangsungkan proses implementasi, penulis dapat langsung mencoba untuk mencari dan menerapkan apa solusi terbaik terhadap masalah tersebut. Untuk proses pengimplementasiannya sendiri dilakukan setelah semua perangkat yang diperlukan berhasil didapatkan, yaitu dimulai dari tanggal 10 November 2014 hingga tanggal 5 Desember 2014, saat proses belajar mengajar semester ini berakhir. Berikut adalah alat-alat pendukung dalam melakukan proses implementasi tersebut.

Hardware :

1. 1 buah alat pembaca *chip* pada kartu mahasiswa (**RFID Card Reader**).
2. 50 buah kartu **RFID** (dengan rincian 42 buah kartu untuk para pelajar dan 8 kartu sebagai kartu).
3. 1 buah komputer dengan spesifikasi
 - i. Prosesor Intel i7 – 3632QM
 - ii. RAM 8 GB
 - iii. *Harddisk* 750 GB
 - iv. *Wireless Mouse*

Software :

1. *Operating Sistem* **Windows 8.1** – *Single Language*
2. Aplikasi *coding* **CodeBlocks** dan **DEV C++**
3. Aplikasi *office* **Microsoft Office 2010**

Sebagai objek Implementasiannya sendiri, berikut adalah tampilan fisik dari alat-alat yang dipakai sebagai objek implementasi tersebut beserta dengan penjelasan mengenai alat-alat yang dipergunakan.

1. **RFID Chip Reader**



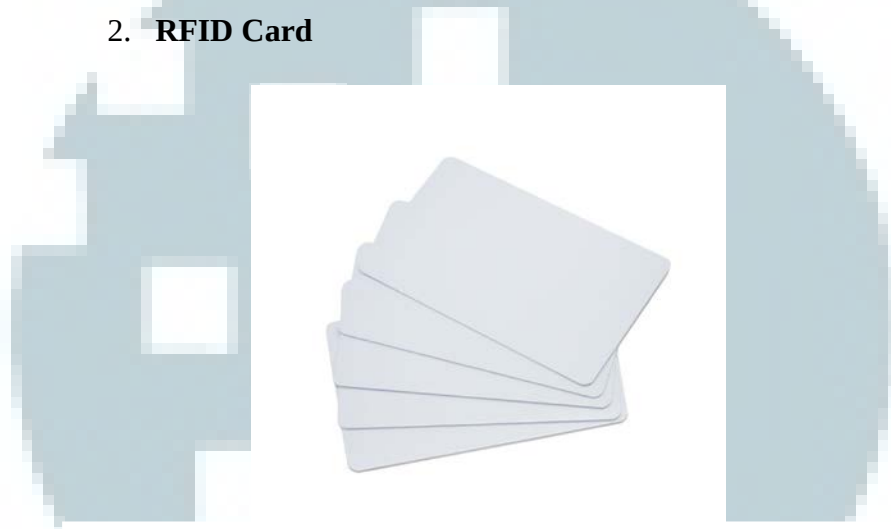
Gambar 3.9. Tampilan RFID Reader

Sedangkan untuk **RFID Chip Reader**, disini kita akan memakai sebuah alat pembaca yang cukup simpel, dimana alat ini hanya dapat membaca isi dari kartu **RFID** tersebut. Alat ini sendiri bekerja dengan menggunakan frekuensi 13,56 mhz. dan digunakan hanya dengan disambungkan ke komputer menggunakan sebuah kabel USB. Alat ini sendiri hanya bisa membaca sebanyak 10 karakter pertama yang tercatat di dalam kartu **RFID**.

Untuk alat yang dipergunakan penulis sendiri dapat dipakai untuk membaca beberapa *chip* **RFID** secara bersamaan sehingga apabila kartu **RFID**

disimpan oleh pelajar di dalam dompet, chip **RFID** yang dicatat di dalam dokumen data pelajar **BIPA UMN** masih dapat terbaca walaupun data nomor kartu dapat berbenturan dengan chip-chip **RFID** lainnya.

2. **RFID Card**



Gambar 3.10. Tampilan Kartu RFID

Kemudian untuk kartu **RFID** yang akan dipergunakan, penulis memilih memakai kartu yang beroperasi di frekuensi 13,56 mhz, sama dengan alat pembaca **RFID** yang penulis miliki. Karena berwujud seperti sebuah kartu putih polos, kartunya dapat saja tertukar satu sama lainnya. Maka, penulis memutuskan untuk menempelkan sebuah *tape* yang berisikan nama panggilan pelajar yang memilikinya, kelas mereka, dan juga program mereka saat ini.

3.12. **Testing Plan**

Untuk proses penerapan sistem yang baru akan dilakukan setelah sebuah proses percobaan singkat terlebih dahulu. Proses *testing* ini dilakukan oleh penulis seorang diri. Dilakukan secara bertahap, pertama penulis akan mencoba

melakukan proses inputing data pelajar baru, lalu mencoba setiap fitur yang dimiliki sistem yang baru, seperti menampilkan daftar pelajar, menambah pengguna aplikasi baru, menampilkan daftar pengguna aplikasi, melakukan proses absensi dari yang harus masuk ke halaman absensi baru melakukan proses *tapping* hingga langsung melakukan proses *tapping* kartu **RFID** dengan **RFID card reader** secara langsung pada halaman utama sistem, dan juga melakukan *testing* pada fitur yang memperbolehkan pelajar untuk melihat daftar kehadiran mereka hingga saat ini.

Setelah proses *testing* yang dilakukan seorang diri tuntas, penulis akan membawa sistem yang baru untuk diimplementasikan dengan **BIPA UMN**. Disana, para pelajar akan dapat melakukan proses *tapping* secara langsung dengan **RFID card reader**. Pada saat yang sama, para pengajar di **BIPA UMN** juga dapat melihat laporan kehadiran pelajar dengan cepat dan juga dapat menambah pelajar-pelajar baru disana. Seluruh fitur dari sistem yang baru akan diujikan disana.

Pada saat implementasi, tidak hanya penulis yang melakukan proses *testing*. Ada beberapa pelajar yang juga datang lalu mencoba untuk melihat daftar kehadiran mereka dengan sistem yang baru. Ada pula pengajar **BIPA UMN** yang mencoba untuk menambah pelajar yang baru dan melihat data kehadiran pelajar. Pada saat implementasi, umumnya para pengajar ingin tahu siapa saja pelajar yang sudah hadir pada hari itu.