

SISTEM INFORMASI NILAI MAHASISWA BERBASIS *MOBILE CLIENT SERVER*

Rika Idmayanti*

*Jurusan Teknologi Informasi, Politeknik Negeri Padang

¹rikaidmayanti@gmail.com

Abstrak

Jurusan Teknologi Informasi Politeknik Negeri Padang dirasa perlu memanfaatkan teknologi *mobile client server* dalam mengelola informasi nilai mahasiswa menggunakan *Java Micro Edition* sebagai *client agent* dan *PHP-MySQL* sebagai *server*. *Client* akan melakukan proses *request data* nilai dan melakukan interaksi dengan *server* melalui koneksi *HTTP (Hyper Text Transfer Protocol)*. Sedangkan *server* merupakan penyedia layanan informasi nilai seperti data mahasiswa, data nilai permata kuliah, data nilai persemester, dan data *IPK* mahasiswa.

Pada penelitian ini aplikasi bekerja dengan beberapa tahap, pertama user diminta mengaktifkan aplikasi Sistem Informasi nilai berbasis *mobile* yang telah diinstal pada *handphone*. Aplikasi membuka koneksi *HTTP* menggunakan *Generic Connection Framework (GCF)*. Untuk membuka jalur koneksi ke *PHP, javax.microedition.io.** menyediakan class *HttpConnection* yang berfungsi untuk membuka jalur koneksi ke alamat yang dituju menggunakan *Connector.open()* melalui jaringan *GPRS*. Kemudian melalui script *PHP* yang ada pada *web server* mengambil data dari *database MySQL* berdasarkan permintaan informasi data nilai dari user. Setelah data diambil oleh aplikasi data akan di-load pada aplikasi berupa *text*.

Sehingga diharapkan mahasiswa dapat dengan mudah mengakses data nilai dimanapun berada, baik secara *mobile* maupun langsung mengakses melalui *web*. Pendekatan penelitian yang penulis lakukan untuk pembuatan aplikasi ini adalah pendekatan metode *Object Oriented Design* menggunakan *Unified Modelling Language*.

Kata Kunci : Sistem Informasi Nilai, Aplikasi Mobile, Client Server, Java 2 Micro Edition, PHP, MySQL.

1. Pendahuluan

Pesatnya perkembangan teknologi *mobile* dan *web* saat ini, menarik perhatian penulis untuk menggabungkan kedua teknologi tersebut dalam sebuah aplikasi pada jurusan Teknologi Informasi Politeknik Negeri Padang dengan merancang dan membuat sebuah sistem informasi nilai mahasiswa berbasis *mobile client server* menggunakan bahasa pemrograman *Java Micro Edition* untuk *client* dan *PHP* dan *MySQL* pada *server*. Aplikasi diharapkan dapat digunakan untuk kebutuhan jurusan Teknologi Informasi dan mahasiswa, mencakup keperluan dalam hal pengaksesan informasi nilai mahasiswa. Mahasiswa bebas untuk mengakses data pada *server* sistem informasi nilai dengan menggunakan *handphone* sebagai *client* dengan syarat aplikasi sudah diinstal pada *handphone* dan layanan *GPRS* pada *handphone* mahasiswa telah diaktifkan.

Dari latar belakang yang telah dijelaskan di atas, maka dapat dirumuskan beberapa masalah yaitu :

- a. Membuat sistem informasi yang mampu menampilkan data nilai permata kuliah, data nilai persemester, data *IPK* dan data mahasiswa.
- b. Merancang sebuah program aplikasi sistem informasi nilai berbasis *mobile* menggunakan

metode *Client Server object oriented design (OOD)* dalam bentuk pembuatan *Unified Modeling Language*.

- c. Membuat program aplikasi *Client-Server* berbasis *mobile* menggunakan *Java 2 Micro Edition* sebagai *client* yang diinstal pada *Handphone* dan *PHP* dan *MySQL* sebagai *Server* (penyedia layanan).

2. Dasar Teori

2.1 Java 2 Platform Micro Edition (JavaME™)

Komponen-komponen *JavaME* terdiri dari *Java Virtual Machine (JVM)* yang digunakan untuk menjalankan aplikasi *Java* pada emulator atau *handled device*, *Java API (Application Programming Interface)* dan *tools* lain untuk pengembangan aplikasi *Java mobile*. Dalam pengembangan aplikasi *wireless*, *JavaME* dibagi menjadi dua bagian yaitu:

1. Lapisan Konfigurasi (*Configuration Layer*)
 - JavaME mempunyai dua konfigurasi yaitu :
 - a. *Connected Limited Device Configuration (CLDC)*
 - b. *Connected Device Configuration (CDC)*.
- Pada tabel 1 menjelaskan perbedaan antara *CDC* dan *CLDC*.

Tabel 1. Perbandingan CDC dan CLDC

CLDC	CDC
Mengimplementasikan sebagian dari J2SE	Mengimplementasikan seluruh fitur J2SE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM
Digunakan pada perangkat genggam (handphone, PDA, <i>twoway pager</i>) dengan memory terbatas (160-512 KB)	Digunakan pada perangkat genggam (internet TV, Nokia Communicator, car TV) dengan memori minimal 2 MB
Prosesor : 16/32 bit	Prosesor : 32 bit

2. Lapisan Profil (*Profile Layer*)

JavaME mempunyai beberapa profil antara lain :

- Mobile Information Device Profile (MIDP)*
- Foundation Profile (FP)*
- Personal Profile*
- Personal Digital Assistance (PDA) Profile*.

2.2 MIDP

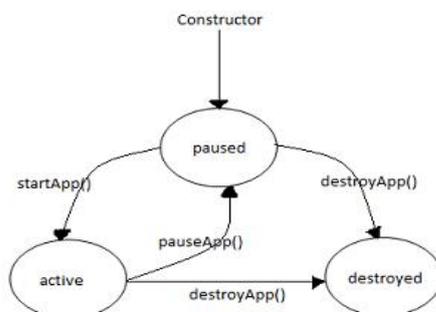
MIDP merupakan *profile* yang banyak digunakan dan populer dari JavaME. Pada Handphone terdapat 2 jenis MIDP, ada yang berupa MIDP 1.0 dan MIDP 2.0. MIDP 1.0 digunakan dalam pemrograman teks, sedangkan MIDP 2.0 digunakan untuk multimedia seperti teks, foto, video, *game* dan animasi.

2.3 MIDlet

MIDlet merupakan aplikasi yang dibuat menggunakan Java 2 Micro Edition dengan *profile Mobile Information Device Profile (MIDP)*.

a. MIDlet Life Cycle

MIDlet *life cycle* merupakan suatu siklus yang menjelaskan bagaimana sebuah MIDlet berfungsi sejak dijalankan sampai dihapus dari memori. Sebuah MIDlet memiliki *life cycle* (daur hidup) sesuai dengan gambar 1.



Gambar 1 MIDlet life cycle

Berdasarkan gambar 2.1, jalannya proses adalah sebagai berikut:

- Sistem akan mencari konstruktor MIDlet.

b. Pada mode *paused*

- Jika terjadi pemanggilan terhadap method `destroyApp()`, aplikasi akan tutup dan sistem memasuki mode *destroyed* atau dihapus dari memori.
- Jika terjadi pemanggilan terhadap method `startApp()`, aplikasi akan mulai dijalankan dan sistem memasuki mode *active*.

c. Pada mode *active*

- Jika terjadi pemanggilan terhadap method `destroyApp()`, aplikasi akan ditutup dan sistem memasuki mode *destroyed* atau dihapus dari memori.
- Jika terjadi pemanggilan terhadap method `pauseApp()`, aplikasi akan dihentikan sementara dan sistem memasuki mode *paused*.

2.4 Method pada aplikasi JavaME

2.4.1 Konstruktor

Sebuah aplikasi JavaME harus memiliki konstruktor, karena disinilah objek-objek yang akan digunakan pada tahap awal harus diinisialisasi. Konstruktor akan di-*load* secara otomatis oleh *device* saat *user* menjalankan aplikasi.

Sebuah konstruktor memiliki nama yang sama dengan nama *class* namun tidak memiliki *return type*. Sebuah aplikasi MIDlet harus didefinisikan sebagai turunan dari class MIDlet bawaan JavaME, sehingga setiap *class* yang dirancang sebagai sebuah MIDlet harus memasang “*extends MIDlet*” di akhir definisinya.

2.4.2 Method Default

Struktur file JavaME memiliki tiga method default yaitu: `startApp()`, `pauseApp()`, dan `destroyApp()`.

- Method `startApp()` dieksekusi saat aplikasi baru berjalan.
- Method `pauseApp()` dipanggil saat aplikasi ingin dihentikan secara sementara.
- Method `destroyApp()` dieksekusi saat aplikasi akan ditutup.

2.5 Generic Connection Framework (GCF)

GCF mendukung koneksi packet (socket) dan stream (*datagram*) berfungsi menyediakan pondasi umum dari berbagai koneksi seperti HTTP. Sebuah GCF memiliki parameter-parameter koneksi yang telah dispesifikasikan dengan menggunakan format pengalamatan:

scheme://username:password@host:port/path;parameters

- scheme* merupakan protokol atau method koneksi. Misalnya: `http,ftp,https`.
- username* bersifat optional, akan tetapi bila ingin didefinisikan harus didahului dengan tanda @

3. *password* bersifat optional dan hanya dapat dispesifikasikan jika *username* telah didefinisikan sebelumnya. Jika *password* didefinisikan, maka harus dipisahkan dari *username* dengan menggunakan titik dua (:)
4. *host* merupakan parameter yang wajib dicantumkan. Bisa berupa nama host atau *fully qualified domain name* (FQDN) atau alamat IP dari host yang dituju.
5. *port* bersifat optional. Jika tidak dispesifikasikan, maka default port akan digunakan
6. *path*
7. parameter bersifat optional, tetapi harus didahului dengan titik koma (;) apabila ia dicantumkan.

Format pengalamatan tersebut haruslah sesuai dengan *Uniform Resource Indicator* (URI) seperti yang didefinisikan pada RFC 2396. Pada MIDP 2.0, hanya skema "http" dan "https" dibutuhkan untuk diimplementasikan pada *device*.

2.6 Membuka Koneksi HTTP

Pada JavaME koneksi HTTP dapat dibuka dengan menggunakan method `Connector.open()` dan meng-casting nya menggunakan salah satu dari ketiga *interface* berikut: `StreamConnection`, `ContentConnection`, dan `HttpConnection`.

```
HttpConnection connection = null;
InputStream iStream = null;
byte[] data = null;
try {
    connection=(HttpConnection)
    Connector.open("http://www.sinilai.ti
    .polinpdg.ac.id/");
    int code
    connection.getResponseCode();
    switch (code) {
    case HttpConnection.HTTP_OK:
    iStream =
    connection.openInputStream();
    int length = (int)
    connection.getLength();
    if (length > 0) {
    data = new byte[length];
    int totalBytes = 0;
    int bytesRead = 0;
    while((totalBytes<length)&&(bytesRead
    > 0)) {
    bytesRead = iStream.read(
    data, totalBytes, length-totalBytes);
    if (bytesRead > 0) {
    totalBytes += bytesRead;
    } }
    } else {
    }break;
    default:
    break;
    }
}
```

Gambar 2 Potongan script membuka koneksi HTTP

2.7 Unified Modeling Language (UML)

UML merupakan bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan disain berorientasi objek. *UML* menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antara *user* dengan *developer*, antara *developer* dengan *developer*, antara *developer* analisis dengan *developer* disain, dan antara *developer* disain dengan *developer* pemrograman. Artifak didalam *UML* didefinisikan sebagai informasi dalam proses pengembangan perangkat lunak. Ada 3 artifak yang penting dari proses analisis dan disain, yaitu *Use Case Diagram*, *Sequence Diagram*, dan *Class Diagram* beserta notasi yang berhubungan dengannya. *Use Case Diagram* merupakan artifak dari proses analisis, sementara *Sequence Diagram* dan *Class Diagram* merupakan artifak dari proses disain.

2.8 PHP

PHP merupakan *server-side embedded script language* artinya sintaks-sintaks dan perintah diberikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

Syntax program/script PHP ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan *tag* PHP yang dapat digunakan untuk menandai blok *script* PHP.

1. `<?php ... ?>`
2. `<script language = "PHP"> ... </script>`
3. `<? ...?>`
4. `<% ... %>`
- 5.

2.9 MySQL

MySQL merupakan turunan salah satu konsep utama dalam *database* yaitu, SQL (*Structured Query Language*). SQL merupakan sebuah konsep pengoperasian *database*, terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

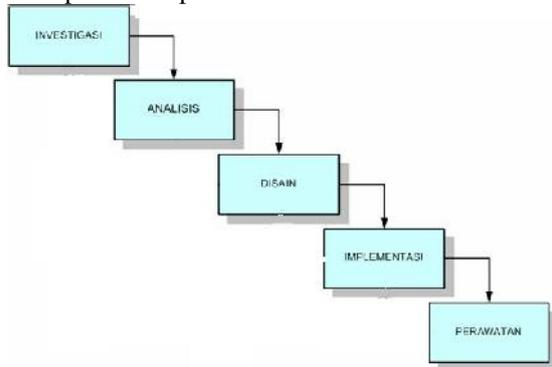
3 Metodologi

Penelitian sistem informasi nilai ini termasuk kedalam penelitian terapan, hal ini disebabkan karena penelitian dilakukan untuk memecahkan suatu permasalahan.

3.1 Tahapan Penelitian

Setiap informasi yang diminta oleh *client* dalam hal ini *mobile/hp* akan disediakan oleh *server*. Supaya penelitian ini menghasilkan output aplikasi seperti yang diharapkan maka ditentukan bahwa metode penelitian yang digunakan adalah metode *waterfall/air terjun*. Metode ini memiliki

fase : investigasi, analisis, desain ;perancangan sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi dan perawatan. Proses ini dapat dilihat pada Gambar 3.



Gambar 3. Metode Waterfall

Output yang dihasilkan pada setiap fase pada metode *Waterfall* adalah:

1. Investigasi, pada fase ini dilakukan konsultasi dengan pengguna sistem. Semua kebutuhan-kebutuhan user didefinisikan secara rinci dan dirumuskan dalam bentuk spesifikasi.
2. Hasil investigasi dianalisis lebih lanjut sehingga didapatkan kebutuhan-kebutuhan yang diperlukan untuk proses perancangan aplikasi sistem informasi nilai. Proses analisis bertujuan memahami masalah yaitu dengan memahami apa yang sebenarnya ada dalam dunia nyata
3. Perancangan Sistem dan Perangkat Lunak, pada fase ini ditentukan arsitektur sistem secara keseluruhan berikut bentuk-bentuk *interface* yang akan dibuat. Perancangan aplikasi yang dilakukan dengan menggunakan pendekatan berorientasi objek. Hasil perancangan ini adalah model sistem dalam bentuk *Unified Modeling Language*
4. Implementasi dan Pengujian Unit, pada fase ini rancangan sistem yang telah dibuat diimplementasikan ke dalam unit-unit program. *Client* dibuat dengan menggunakan teknologi Java MicroEdition. *Client* kemudian diinstal ke *mobile/Handphone user*. Sedangkan Server berupa aplikasi berbasis web yang diimplementasikan dengan menggunakan bahasa pemrograman PHP dan MySQL, Sehingga diketahui dan dapat dipastikan apakah masing-masing unit aplikasi pada *client* dengan *server* sudah bekerja dengan baik.
5. Integrasi dan Pengujian Sistem, pada fase ini unit-unit sudah terpenuhi oleh sistem. Integrasi dan Pengujian Sistem dilakukan dengan menginstal aplikasi *Client* di

Handphone dan dilakukan pengujian interkoneksi dengan Server yang sebelumnya sudah di *hosting/online*. Pada fase ini diketahui dan dapat dipastikan apakah antara *client* dengan *server* dapat berinteraksi dan bertukar data. sesuai dengan informasi yang diminta oleh *user*.

6. Perawatan, pada fase ini aplikasi *client* yang sudah jadi diinstal pada *handphone* mahasiswa dengan mengirimkan file jar aplikasi ke *handphone*. Proses pemeliharannya adalah mengecek sistem yang ada pada user dan dilakukan pengujian untuk sistem untuk mengecek apakah semua persyaratan yang diminta oleh *user* telah terpenuhi, apakah masih terjadi error atau tidak. Bila masih terjadi error maka sistem diperbaiki sampai terbebas dari kesalahan. Namun sebelumnya sudah dilakukan pelatihan penggunaan program aplikasi yang dibuat kepada user.

3.2 Lokasi Penelitian

Penelitian secara umum dilaksanakan pada Jurusan Teknologi Informasi Politeknik Negeri Padang Sumatera Barat.

3.3 Peubah Penelitian

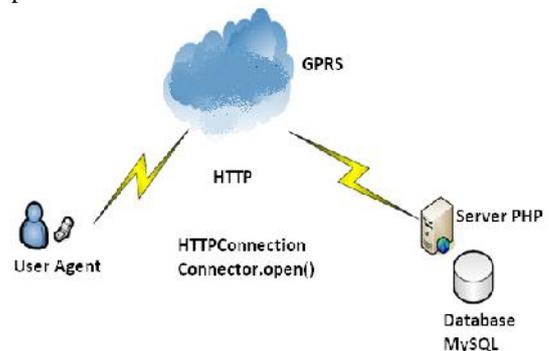
Penelitian ini adalah penelitian terapan dengan data peubah penelitian adalah :

- Data Mahasiswa
- Data nilai permata kuliah
- Data nilai yudisium kelulusan mahasiswa
- Data index prestasi mahasiswa

4 Pembahasan

4.1 Membuka Koneksi HTTP Antara *Client* dan *Server*

Penelitian aplikasi Sistem Informasi Nilai *Mobile Client Server* ini memiliki gambaran seperti Gambar 4.

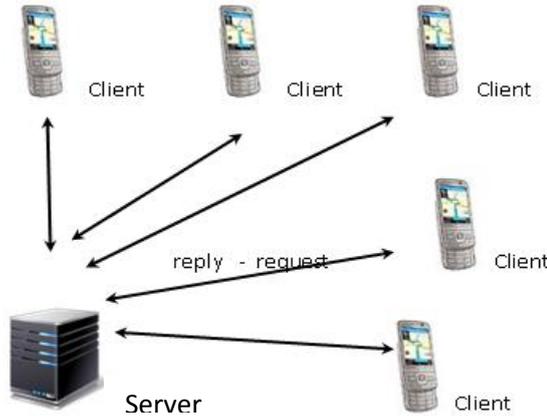


Gambar 4. Aplikasi Mobile Client Server

Urutan kerja aplikasi sebagai berikut :

1. Pengguna mengaktifkan aplikasi Sistem Informasi nilai berbasis *mobile* yang telah diinstal pada *handphone*.

2. Setelah itu aplikasi membuka *HttpConnection* menggunakan *Connector.open()* ke *web server* melalui jaringan GPRS.
3. Kemudian melalui script PHP yang ada pada web server mengambil data dari database MySQL berdasarkan permintaan data dari user.
4. Setelah data diambil oleh aplikasi data akan *diload* pada aplikasi berupa text String seperti Gambar 5.



Gambar 5. Proses Reply Request Server Client

Pengiriman pesan oleh user di awali dengan pengiriman perintah ke server HTTP. Pada spesifikasinya HTTP mendukung perintah HEAD, GET, POST, PUT, DELETE, dan sebagainya. Perintah HTTP yang dipergunakan untuk aplikasi pembacaan halaman web serta pengiriman nilai untuk di proses oleh teknologi *Server-side Scripting* hanya menggunakan perintah GET dan POST saja. Perbedaan antara GET dan POST adalah bahwa pada penggunaan GET, nilai variabel dikirimkan dengan menuliskannya pada URL sehingga jumlahnya terbatas, Sedangkan nilai variabel pada POST dituliskan pada *body* pesan.

Pada method GET, parameter telah dispesifikasikan dan dilekatkan pada URL. Sebagai contoh, untuk memberikan sebuah *variable* dengan nama “nobp” dan memiliki nilai 1409050 kepada halaman *index.php*, URL tersebut akan dispesifikasikan sebagai : “*http://hostname/index.php?nobp=1409050*”. Parameter tambahan dipisahkan dengan dengan tanda &, “*http://hostname/index.php?nobp=1409050&page=2*”. Jika POST digunakan, parameter bukanlah menjadi bagian dari URL tetapi dikirim dengan pada baris terpisah pada POST. Pada penelitian ini *method* yang penulis gunakan adalah GET.

4.2 Paradigma Object Oriented

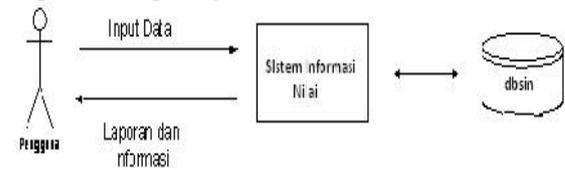
Metoda analisis dan perancangan sistem dengan paradigma berorientasi objek memiliki ciri : Fokus pada objek, data dan prosedur disimpan dalam objek dan dekomposisi data.

Aplikasi *Object Oriented* sangat cocok untuk dikembangkan pada situasi pembuatan program yang sangat memperhatikan evolusi/pengembangan dari kemampuan aplikasinya. Paradigma perancangan Pemrograman berbasis *Object Oriented Design(OOD)* menggunakan pendekatan UML(*Unified Modeling Language*).

4.3 Unified Modeling Language

4.3.1 Gambaran Umum Sistem

Perangkat lunak yang dibangun bernama sistem informasi nilai berbasis *mobile client server* untuk pengelolaan nilai pada Jurusan Teknologi Informasi Politeknik Negeri Padang. Gambaran umum sistem informasi nilai dapat dilihat pada gambar 6.



Gambar 6. Gambaran Umum Sistem

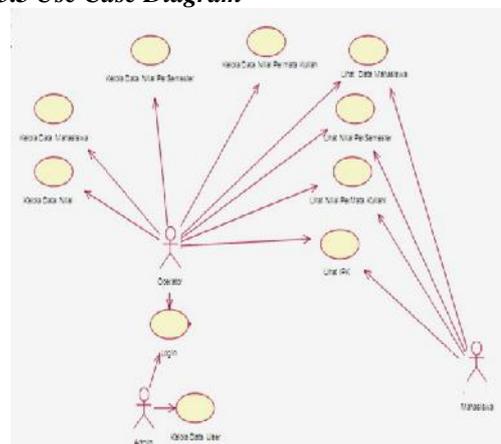
Pengguna sistem ini admin, mahasiswa dan dosen yang berwenang di Jurusan Teknologi Informasi untuk menginputkan data-data yang berhubungan dengan data mahasiswa dan data nilai.

4.3.2 Karakteristik Pengguna

Tabel 2. Karakteristik Pengguna

Aktor	Deskripsi
Administrator	Menambah dan Menghapus User
Operator	Menginput data dosen, data nilai, data nilai persemester, data nilai permatakuliah, dan data nilai IPK
Mahasiswa	Melihat data nilai, data jadwal, data nilai persemester, data nilai permatakuliah, dan data nilai IPK

4.3.3 Use Case Diagram



Gambar 7. Use Case Diagram Sistem

4.3.3 Definisi Use Case Diagram

Identifikasi pemodelan *use case diagram* sesuai gambar 4.4 dapat dilihat pada Tabel 3:

Tabel 3. Defenisi Use Case

Use Case	Deskripsi
Login	Admin, operator dan pimpinan melakukan <i>login</i>
Kelola Data Mahasiswa	Operator dapat menambah, mengubah, menghapus, mencari, dan menampilkan data mahasiswa
Kelola Data Nilai Persemester	Operator dapat menambah, mengubah, menghapus, mencari, dan menampilkan data nilai persemester
Kelola Data Nilai Per Matakuliah	Operator dapat menambah, mengubah, menghapus, mencari, dan menampilkan data nilai permatakuliah
Lihat Data Mahasiswa	Mahasiswa melalui client/HP dapat melihat data mahasiswa
Lihat Nilai Persemester	Mahasiswa melalui client/HP dapat melihat data nilai persemester
Lihat Nilai IPK	Mahasiswa melalui client/HP dapat melihat data nilai persemester

1. Contoh Skenario *Use Case* Kelola Data Nilai Per Matakuliah

Use Case : Kelola Data Nilai Per Mata kuliah
 Aktor : Operator
 Tujuan : Melihat, mengakses, mengelola data nilai per matakuliah

Tabel 4. Skenario Use Case Kelola Data Nilai Per Matakuliah

Aksi Aktor	Reaksi Sistem
Skenario Normal	
Operator mengelola data nilai per matakuliah	Sistem menerima pengelolaan data mulai dari tambah, cari, ubah, hapus, dan menampilkan data nilai per matakuliah
Skenario Salah	
Pengguna memasukkan kode matakuliah salah	Sistem tidak dapat menampilkan data nilai per matakuliah sesuai dengan kode matakuliah yang diinputkan dan menampilkan pesan kesalahan

2. Contoh Skenario *Use Case* Lihat Nilai Persemester

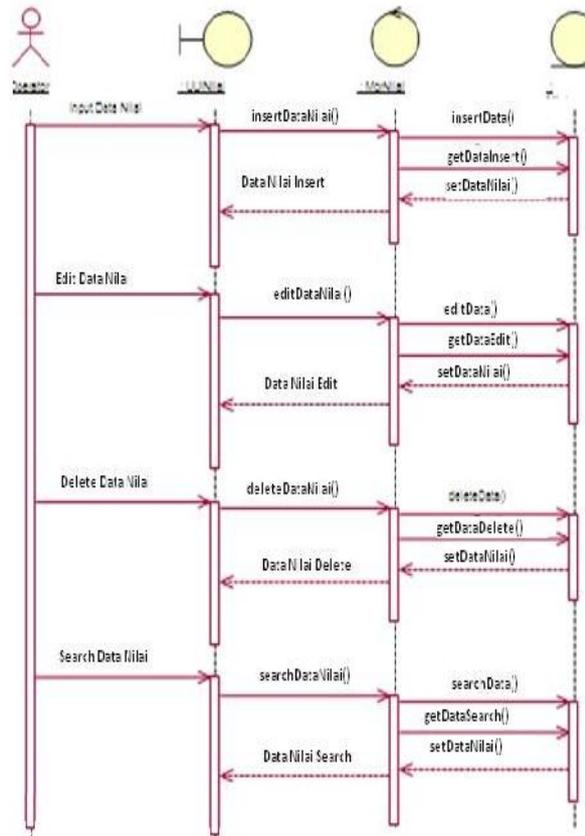
Use Case : Lihat Nilai Persemester
 Aktor : Mahasiswa
 Tujuan : Melihat dan mengakses data nilai persemester

Tabel 5. Skenario Use Case Lihat Data Nilai Persemester

Aksi Aktor	Reaksi Sistem
Skenario Normal	
Mahasiswa melalui client/HP dapat melihat data nilai IP sesuai dengan semester yang diinginkan	Aplikasi membalas permintaan data Semua data nilai yang telah ditempuh aktor sesuai

	dengan semester yang diinputkan
Skenario Salah	
Pengguna memasukkan <i>keyword</i> semester yang salah	Sistem tidak dapat memberikan jawaban data nilai persemester dan menampilkan peringatan bahwa terjadi kesalahan dalam memasukkan <i>keyword</i> semester

4.3.4 Sequence Diagram Sistem Informasi



Gambar 8. Sequence Diagram Sistem Informasi Nilai

4.4 Perancangan Antar Muka

a. Contoh Perancangan Input Melihat Nilai Persemester

Desain proses melihat nilai per semester dapat dilihat pada gambar 9.

Nilai Persemester

Bp

Tahun Ajaran

Prodi ▼

Kelas ▼

Semester ▼



Gambar 9. Proses Melihat Nilai Per Semester

b. Contoh Perancangan Desain Output Data Nilai Persemester

Desain output digunakan untuk merancang tampilan data yang dihasilkan dari hasil input oleh pemakai sistem.



Gambar 10. Desain Ouput Data Nilai Persemester

4.5 Hasil Dan Pembahasan

4.5.1 Implementasi Pemanfaatan Hardware dan Software

Dalam pembuatan dan pengembangan sistem informasi ini serta dalam pembuatan laporannya memanfaatkan spesifikasi hardware dan software. Dalam pembuatan Antar muka client penulis NetBeans Integrated Development Environment (IDE). Netbeans berjalan pada sistem operasi yang mendukung Java Virtual Machine dan Java Runtime Environment. Berikut ini merupakan daftar dari platform-platform sistem operasi yang bisa digunakan:

1. Microsoft Windows XP Profesional SP3/Windows Vista/Windows 7
2. Ubuntu 9.04
3. Solaris versi 10 (SPARC)
4. Solaris versi 10 (x86/x64/ edisi platform)

5. Macintosh OS X 10.5 Intel
6. Macintosh OS X 10.5 PPC

4.5.2 Koneksi dari Client ke Server

Pada server dibuatkan fungsi-fungsi yang menangani request dari client dan sekaligus memastikan apakah request yang dikirimkan client tersedia datanya pada server. Jika data yang di request ada, maka server akan memberikan balasan sesuai dengan request yang dikirimkan oleh client. Aplikasi sistem informasi nilai menggunakan thread yang mengimplementasikan interface Runnable.

4.5.3 Tampilan Implementasi Antar Muka Server Sistem Informasi Nilai



Gambar 11. Tampilan Antar Muka Server

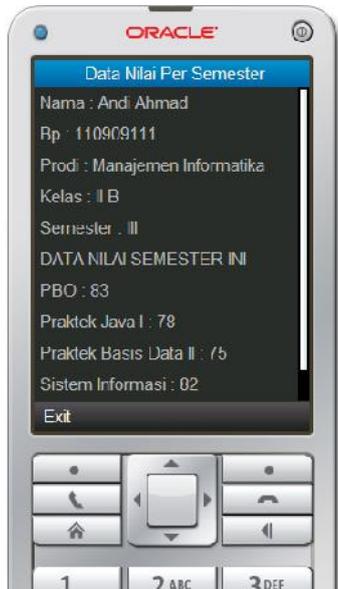
4.5.4 Tampilan Implementasi Antar Muka Client pada Emulator



Gambar 12. Tampilan Splash Screen Emulator Client

Contoh Tampilan Output Nilai per Semester

Gambar 13 menampilkan tampilan *emulator reply server* nilai per semester yang di *request* mahasiswa pada *mobile client*



Gambar 13. Output Nilai per Semester

Jika data nilai persemester mahasiswa yang di-*request* ke *server* sistem informasi tidak ditemukan maka *response* yang dikirimkan oleh *server* kepada *client* dapat dilihat pada gambar 14.



Gambar 14. Response dari Server jika Data tidak ditemukan

5. Kesimpulan

Berdasarkan pengamatan dan analisa selama melakukan pengumpulan data pada Jurusan Teknologi Informasi Politeknik Negeri Padang, maka dapat diambil beberapa kesimpulan diantaranya :

1. Melihat banyaknya data nilai di Jurusan Teknologi Informasi Politeknik Negeri

Padang, menjadikan kebutuhan akan informasi menjadi meningkat, khususnya dalam hal pengolahan data nilai mahasiswa.

2. Dengan memanfaatkan sistem informasi nilai berbasis *mobile client server* data yang dibutuhkan dapat diakses kapanpun dan dimanapun oleh seluruh Staf dan Mahasiswa TI lewat *website* maupun *handphone*.
3. Proses pencarian, pemasukan, serta pengeditan data nilai di jurusan dapat dilakukan dengan mudah.

6. Daftar Pustaka

- Hermawan, Julius. 2004. *Analisa Disain dan Pemograman Obyek dengan UML dan Visual Basic.Net*, Edisi I. Yogyakarta: Andi.
- Irwanto, Djon. 2007. *Membangun Object Oriented Software dengan Java dan Object Database*. Jakarta: Elex Media Komputindo.
- Jogiyanto, HM. 2005. *Analisa & Disain : Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta : Andi,.
- Paranginangin, Kasiman. 2007. *Aplikasi Web dengan PHP dan MySQL*. Jakarta : Andi.
- Purnama,Rangsang. 2005. *Mari Mengenal J2ME*. Jakarta : Prestasi Pustaka Publisher.
- Shalahuddin & Rossa A.S. 2010. *Pemrograman JavaME* .
- Sommerville, Ian . 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga