

**m-Simpol**

**TUGAS AKHIR**

Oleh :

**Muhamad Triyadi 3310701100**

**Sigit Febrianto 3310801093**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BATAM**

**BATAM**

**2011**

# **LEMBAR PENGESAHAN**

Batam, Agustus 2011

**Pembimbing,**

**Mir'atul Khusna Mufida S.ST**

**NIK.**

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310701100

Nama : Muhammad Triyadi

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

### **m-Simpol**

disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, Agustus 2011

**Muhamad Triyadi**  
3310701100

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801093

Nama : Sigit Febrianto

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

### **m-Simpol**

disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, Agustus 2011

**Sigit Febrianto**  
3310801093

## HALAMAN PERSEMBAHAN

**M. Triyadi**

↳ Kepada Allah,,

Zat tunggal penguasa isi alam semesta dan segala pergerakannya yang penuh kejutan. Terima kasih untuk segala rupa dalam hidupku. Kau Maha Segalanya.

↳ Kepada Bapak dan Mama,,

Semua perjalanan hidup tentunya bermula dan akan selalu disyukuri atas bapak dan mamaku (Supardi BN & Nyanyu Gita Dewiyani) yang Amat disayangi. Karena mereka senyum ini hidup dan tangis ini berarti dalam karya ini. Terima kasih untuk segala percaya yang tak ada habisnya.

↳ Kepada Saudara Sedarahku,,

Buat adiku (Rizka Nursandi) semoga kelak bisa menjadi orang yang berbakti kepada orang tua. Dan abangku (Tedy Ardiansyah) semoga memiliki keluarga syakinah mawadah warahmah yang telah memasuki kehidupan baru. Dan untuk kakaku (Ayu Agustin) semoga kelak bisa menjadi orang yang berbakti kepada orang tua dan berguna.

↳ Kepada Sahabat,,

Buat My Partner Tugas Akhir ini (Dewa Puja Pratama) yang selalu memberi supportnya, Para pemberi semangat tanpa jemu Farah, Anto, Fita, Tika, Sarah, Fitra, Prima, Arman, Mariaty, Ipit, Ferri, Lis, Fiza, Afriz. Terima kasih untuk hari-hari aneka rasa.

↳ Kepada Dosen,,

Manusia-manusia terpilih tanpa pamrih Bpk.Riwinoto, Bpk. Uuf Bradawidagda, Bpk. Toar, Bpk. Afdhol Zikri, Ibu Riska, dll. Terima kasih atas semua jawaban akan tanda tanyaku. Atas semua ikhlas yang tak bisa kubalas.

↳ Kepada Kamu,,

Yang selalu memberiku semangat untuk menyelesaikan Tugas Akhir ini. Terima kasih atas segala perhatianmu. Serta semua jiwa yang singgah dalam hidupku.

**HALAMAN PERSEMBAHAN**

**Sigit Febrianto**

## KATA PENGANTAR

Segala puji dan syukur kita panjatkan kepada Allah SWT yang telah memberikan Rahmat dan Karunia-Nya, penulis bersyukur kepada kepada Allah SWT atas selesainya Tugas Akhir yang berjudul "m-Simpol"

Dalam kesempatan ini pula penulis mengucapkan terima kasih kepada:

1. Allah SWT atas terselesainya Tugas Akhir ini.
2. Kedua orang tua yang telah memberikan dukungan moril maupun materil
3. Bapak DR.Ir.Priyono Eko Sanyoto, M.Sc, selaku Direktur Politeknik Negeri Batam
4. Bapak Uuf Brajawidagda selaku koordinator Tugas Akhir
5. Ibu Mir'atul Khusna Mufida selaku dosen Pembimbing Tugas Akhir ini. Terima kasih atas bimbingan dan motivasi yang bapak berikan dalam penyelesaian laporan ini.
6. Keluarga yang telah memberikan do'a, dukungan dan semangat.
7. Bapak/ Ibu dosen program studi Teknik Informatika Politeknik Negeri Batam atas arahan dan bimbingannya.
8. Kepada widodo hidayat yang selalu memberi dukungan serta membantu kami disaat kami mengalami kesulitan.
9. Teman-teman Teknik Informatika angkatan 2008 yang memberikan dukungan, semangat dan do'a.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan ini. Oleh karena itu, penulis sangat mengharapkan bantuan dari beberapa pihak baik berupa kritik maupun saran yang membangun dari pembaca. Akhir kata penulis mengucapkan terima kasih, semoga penulisan laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah penelitian yang serupa.

Batam, Agustus 2011

Penulis

## ABSTRAK

### m-Simpol

Sistem Informasi Mahasiswa Politeknik Negeri Batam (SIMPOL) adalah sistem informasi yang memfasilitasi mahasiswa untuk mendapatkan layanan akademik, SIMPOL saat ini hanya dapat diakses menggunakan komputer dan ruang lingkup penggunaan SIMPOL tidak dapat diakses di luar lingkungan Politeknik Negeri Batam, karena sistem yang ada saat ini hanya dapat diakses melalui *local area network (LAN)*. Mengingat pentingnya SIMPOL mengakibatkan adanya permasalahan yang timbul oleh kelalaian pengguna SIMPOL misalnya, mahasiswa yang sering terlambat daftar ulang dan keterbatasan media yang ada.

Perkembangan teknologi *mobile phone* yang didukung oleh internet, menciptakan akses data di setiap waktu dan tempat. *Mobile phone* menjadi teknologi yang maju dan dibutuhkan pada setiap bidang pekerjaan. Pada umumnya saat ini masyarakat lebih banyak menggunakan *mobile phone* dibandingkan dengan *notebook*, Oleh karena itu dibuatlah aplikasi m-SIMPOL yang bertujuan mempermudah mahasiswa untuk dapat mengakses layanan akademik.

Dengan adanya aplikasi m-SIMPOL, membuat mahasiswa dan dosen lebih mudah mengakses layanan akademik seperti daftar ulang dan pengolahan nilai. Aplikasi m-SIMPOL menyediakan layanan akademik melalui media *mobile phone*, yang dapat diakses di luar lingkungan Politeknik Negeri Batam.

Kata Kunci: SIMPOL, *mobile phone*, m-Simpol

# **ABSTRACT**

## **m-Simpol**

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN .....	ii
LEMBAR PERSEMBAHAN .....	iv
KATA PENGANTAR .....	vi
ABSTRAKSI .....	vii
ABSTRACT .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xiv
Bab I Pendahuluan.....	16
I.1 Latar Belakang.....	16
I.2 Rumusan Masalah.....	17
I.3 Batasan Masalah .....	17
I.4 Tujuan .....	17
Bab II Landasan Teori .....	18
II.1 Simpol.....	18
II.1.1 Daftar Ulang .....	18
II.1.2 Biaya Persemester.....	19
II.1.3 IP.....	19
II.1.4 IPK.....	20
II.1.5 Melihat Nilai.....	20
II.2 Pemrograman Java.....	20
II.2.1 Sifat-sifat Pemrograman Java.....	21
II.2.2 Pemrograman J2ME .....	22
II.3 Unified Modelling Language (UML).....	25
II.3.1 Konsepsi Dasar UML .....	27
II.3.2 Use Case Diagram .....	29
II.3.3 Kelas Analisis .....	31
II.3.4 Diagram Sekuen .....	33

II.3.5	Diagram Kelas .....	33
II.4	NetBeans.....	35
II.4.1	Platform NetBeans.....	35
II.4.2	NetBeans IDE .....	36
Bab III	Analisis dan Perancangan.....	41
III.1	Diskripsi Umum Aplikasi m-Simpol.....	41
III.1	Kategori pengguna aplikasi .....	42
III.2	Fitur Utama Perangkat Lunak.....	43
III.3	Kebutuhan Fungsional.....	43
III.4	Batasan Sistem.....	43
III.5	Use Case .....	44
III.5.1	Diagram Use Case .....	44
III.5.2	Skenario Use Case .....	44
III.6	Analisis Kelas .....	48
Bab IV	Perancangan.....	49
IV.1	Squence Diagram.....	49
IV.2	Diagram Kelas .....	52
IV.2.1	Gambar Digram Kelas m-Simpol.....	52
IV.3	Rancangan Kelas Rinci.....	53
IV.3.1	Kelas Mobile device .....	53
IV.3.2	Kelas Koneksi Mobile .....	53
IV.3.3	Kelas Koneksi Server .....	53
IV.3.4	Kelas Mahasiswa .....	<b>Error! Bookmark not defined.</b>
IV.3.5	Kelas Nilai .....	<b>Error! Bookmark not defined.</b>
IV.3.6	Kelas Matakuliah.....	<b>Error! Bookmark not defined.</b>
IV.3.7	Kelas Daftar Ulang .....	<b>Error! Bookmark not defined.</b>
IV.4	Perancangan Antar Muka .....	54
IV.4.1	Login dosen .....	54
IV.4.2	Login Mahasiswa.....	55
IV.4.3	Input Nilai.....	56
IV.4.4	Update Nilai.....	59

IV.4.5	Lihat Nilai .....	61
IV.4.6	Daftar Ulang .....	62
Bab V	Daftar Pustaka.....	65

## DAFTAR GAMBAR

Gambar 1.	Tampilan Simpol.....	18
Gambar 2.	Daur hidup MIDlet.....	25
Gambar 3.	Actor.....	30
Gambar 4.	Use Case.....	31
Gambar 5.	Use Case Diagram.....	31
Gambar 6.	Kelas Analisis .....	32
Gambar 7.	Diagram Sekuen.....	33
Gambar 8.	Diagram Kelas.....	34
Gambar 9.	Toolbar pada NetBeans.....	36
Gambar 10.	Tampilan Choose Project.....	37
Gambar 11.	Tampilan Name and Location .....	37
Gambar 12.	Tampilan Default Platform Selection .....	38
Gambar 13.	Source code pada J2ME .....	39
Gambar 14.	Source code pada J2ME .....	39
Gambar 15.	Tampilan Emulator .....	40
Gambar 16.	Deskripsi Umum Sistem .....	41
Gambar 17.	Diagram Use Case Aplikasi m-Simpol.....	44
Gambar 18.	Analisis Kelas .....	48
Gambar 19.	Squence Diagram Use Case Input Nilai .....	49
Gambar 20.	Squence Diagram Use Case Update Nilai .....	50
Gambar 21.	Squence Diagram Use Case Lihat Nilai.....	51
Gambar 22.	Squence Diagram Use Case Daftar Ulang.....	51
Gambar 23.	Squence Diagram Use Case Verifikasi.....	52
Gambar 24.	Squence Diagram Use Case Drop Matakuliah.....	52
Gambar 25.	Diagram kelas.....	53
Gambar 26.	Tampilan Login Dosen.....	55
Gambar 27.	Tampilan Login Mahasiswa.....	56
Gambar 28.	Tampilan Input Nilai.....	57
Gambar 29.	Tampilan Update Nilai.....	59

Gambar 30.	Tampilan Lihat Nilai.....	61
Gambar 31.	Tampilan Daftar Ulang .....	62
Gambar 32.	Skenario Pengujian m-SIMPOL.....	64

## DAFTAR TABEL

Tabel 1.	Bobot Nilai Indeks Prestasi.....	19
Tabel 2.	Konsep Dasar UML.....	27
Tabel 3.	Kardinalitas.....	24
Tabel 4.	Tingkatan Pengguna Aplikasi.....	42
Tabel 5.	Tabel Kelas Mobile Device.....	53
Tabel 6.	Tabel Koneksi Mobile.....	53
Tabel 7.	Tabel Kelas Koneksi Server.....	54
Tabel 8.	Tabel Implementasi Kelas.....	62
Tabel 9.	Tabel Implementasi Antar Muka.....	62
Tabel 10.	Hasil Rinci Pengujian.....	65

## Bab I Pendahuluan

### I.1 Latar Belakang

Sistem Informasi Mahasiswa Politeknik Negeri Batam (SIMPOL) adalah sistem informasi yang memfasilitasi mahasiswa untuk mendapatkan layanan akademik, seperti melihat nilai, melakukan daftar ulang, jumlah pembayaran persemester, perwalian dan melihat *indeks prestasi (IP) / indeks prestasi kumulatif (IPK)*. SIMPOL saat ini hanya dapat diakses menggunakan komputer, dan ruang lingkup penggunaan SIMPOL tidak dapat diakses di luar lingkungan Politeknik Negeri Batam, karena sistem yang ada saat ini hanya dapat diakses melalui *local area network (LAN)*. Mengingat pentingnya SIMPOL mengakibatkan adanya permasalahan yang timbul oleh kelalaian pengguna SIMPOL. Permasalahan – permasalahan yang ditimbulkan misalnya, mahasiswa yang sering terlambat daftar ulang, kesulitan dosen dalam memasukan nilai serta keterbatasan media (Tidak semua mahasiswa memiliki komputer/*notebook*).

Kebutuhan manusia yang beragam mengakibatkan teknologi informasi terus maju dan berkembang, salah satu kebutuhan pokok manusia ialah mereka ingin segala pekerjaan dan urusannya dapat diselesaikan dengan cepat dan efisien, sehingga tidak membuang banyak waktu dan tenaga. Kebutuhan inilah yang membuat teknologi *mobile phone* semakin berkembang.

Perkembangan teknologi *mobile phone* yang didukung oleh internet, menciptakan akses di setiap waktu dan tempat, dikarenakan terciptanya mikroprosesor berukuran kecil dan efektif yang mempunyai kecepatan tinggi. *Mobile phone* menjadi teknologi yang maju dan dibutuhkan pada setiap bidang pekerjaan. Pada umumnya saat ini masyarakat lebih banyak menggunakan *mobile phone* dibandingkan dengan *notebook*, dengan begitu penggunaan *mobile phone* lebih cocok untuk mengakses informasi.

Berdasarkan masalah yang dihadapi diatas, maka diperlukan suatu bentuk layanan informasi akademik yang dapat mempermudah mengakses data

SIMPOL sebagai server. Karena itu dibutuhkan aplikasi *mobile* simpol (m-SIMPOL), yang bertujuan mempermudah mahasiswa untuk dapat mengakses layanan akademik dari data server melalui *client* yang dibuat menggunakan teknologi *mobile*.

## **I.2 Rumusan Masalah**

Rumusan masalah dari tugas akhir ini

1. Bagaimana mengakses data SIMPOL pada *mobile phone*.
2. Bagaimana mengakses SIMPOL di luar lingkungan Politeknik Negeri Batam.

## **I.3 Batasan Masalah**

1. Aplikasi ini hanya menangani layanan dosen pengampu, dosen wali dan mahasiswa.
2. Aplikasi ini tidak menangani keamanan jaringan.
3. Aplikasi ini tidak menangani aplikasi server melalui j2me.
4. Aplikasi ini hanya dapat diakses melalui *mobile phone* dengan spesifikasi yang mendukung konfigurasi dan profile dari J2ME.

## **I.4 Tujuan**

Tujuan dari tugas akhir ini adalah :

1. Menyediakan layanan akademik mahasiswa melalui *mobile phone*, yang dapat mengakses data *server* melalui *mobile phone*.
2. Menyediakan layanan akademik mahasiswa yang dapat diakses di luar lingkungan Politeknik Negeri Batam.

## Bab II Landasan Teori

Bagian ini memuat teori-teori yang berhubungan dengan penelitian. Juga dapat diulas penelitian-penelitian bidang sejenis yang pernah dilakukan serta posisi penelitian tersebut terhadap penelitian sebelumnya. Bagian ini mencerminkan penguasaan materi oleh penulis TA. Berikan pengantar tentang isi bab ini di sini.

### II.1 Simpol

Sistem informasi mahasiswa Politeknik Negeri Batam adalah sistem informasi yang memfasilitasi mahasiswa untuk mendapatkan layanan akademik, seperti melihat nilai, melakukan daftar ulang, jumlah pembayaran persemester, konseling dan melihat IP/IPK.

SISTEM INFORMASI AKADEMIK POLITEKNIK BATAM  
 NIM : 3310701117  
 NAMA : Danny Rio Setiawan  
 TEMPAT LAHIR : Boyolali  
 TANGGAL LAHIR : 1988-12-03  
 ALAMAT : Bengkong Harapan 1 Blok F No.53 RT.05 RW.18 Batam

ITEM	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 6	Sem 7	Sem 8	Sem 9
A. SKS DIAMBIL	22	21	12	17	19	19	14	20	
B. TOT SKS DIAMBIL	22	43	55	72	91	110	124	144	
C. SKS->TRANSKRIP	22	9	2	14	19	19	14		
D. TOT SKS->TRANSKRIP	22	31	33	47	66	85	99	99	
E. BIAYA HARUS DIBAYAR	5.540.000	1.695.000	1.290.000	1.515.000	1.605.000	1.605.000	1.380.000	1.650.000	
F. BEASISWA	0	0	0	0	0	0	0	0	
G. TOTAL DIBAYARKAN	5.540.000	1.695.000	1.290.000	1.515.000	1.605.000	1.605.000	1.380.000	1.650.000	
H. SISA = F + G - E	0	0	0	0	0	0	0	0	
I. IP	2.23	0.81	1.5	2.12	1.74	3	4		
J. IPK	2.23	2.06	2.18	2.3	2.14	2.33	2.57		
K. KETIDAKHADIRAN									
L. DAFTAR ULANG									sem 9
M. PERWALIAN								sem 8	
N. GANTI PASSWORD									

Gambar 1. Tampilan Simpol

#### II.1.1 Daftar Ulang

Daftar ulang adalah salah satu syarat untuk mahasiswa bisa melanjutkan ke semester berikutnya. Dimana mahasiswa diwajibkan untuk mengambil mata kuliah dengan cara daftar ulang.

Mahasiswa yang boleh mengambil mata kuliah adalah mahasiswa yang telah memenuhi ketentuan pengambilan mata kuliah, Jika mahasiswa telah memenuhi ketentuan yang ada maka mahasiswa harus menghadap Wali Akademik untuk mengetahui mata kuliah yang bisa diambil oleh mahasiswa sesuai jurusannya, setelah mahasiswa melakukan pengambilan mata kuliah,

Wali Akademik akan melakukan verifikasi, kemudian mahasiswa membayar uang kuliah sesuai dengan biaya yang telah ditetapkan.

Bagi mahasiswa yang ingin melakukan penundaan pembayaran, mahasiswa harus mengisi borang penundaan pembayaran dan harus disetujui serta ditangani oleh Wali Akademik.

### II.1.2 Biaya Persemester

Jumlah yang harus dibayar oleh mahasiswa pada semester tertentu. Jumlah total yang harus dibayar oleh seorang mahasiswa pada semester tertentu. Terdiri dari BPP, SPP dan BSKS. Tiap mahasiswa mungkin berbeda.

### II.1.3 IP

Indeks Prestasi adalah Penilaian keberhasilan seorang mahasiswa yang dinyatakan dengan nilai kredit rata-rata yang merupakan satuan nilai akhir yang menggambarkan mutu penyelesaian suatu program studi. Indeks Prestasi dihitung pada setiap akhir semester. Indeks Prestasi mahasiswa dihitung berdasarkan pembobotan pada Indeks Nilai sebagai berikut :

**Tabel 1. Bobot Nilai Indeks Prestasi**

Indeks	Bobot
A	4
B	3
C	2
D	1
E	0

Indeks Prestasi Semester (IPS) , dihitung dari nilai mata kuliah untuk semua mata kuliah yang diambil pada satu semester, dengan diberi bobot SKS untuk tiap mata kuliah yang bersangkutan.

#### II.1.4 IPK

Indeks Prestasi Kumulatif (IPK) merupakan angka yang menunjukkan prestasi atau kemajuan belajar mahasiswa secara kumulatif mulai dari semester pertama sampai dengan semester paling akhir yang telah ditempuh.

Rumus perhitungannya sebagai berikut (pembulatan ke bawah apabila kurang dari 0,05, pembulatan ke atas apabila sama/lebih dari 0,05) :

$$\text{IPK} = \frac{\text{Jmlh ( AM x SKS ) seluruh semester yang ditempuh}}{\text{Jmlh SKS seluruh semester yang ditempuh}}$$

#### II.1.5 Melihat Nilai

Berisi tentang mata kuliah yang diambil sampai dengan sebuah semester (sesuai dengan nomor semester pada kolom tersebut). Semua sks dihitung, termasuk yang bernilai E atau belum keluar nilainya. Dan yang disetujui wali akademik untuk masuk transkrip. Nilai 0 dan E tidak akan dihitung (masuk ke dalam transkrip)

### II.2 Pemrograman Java

“Program adalah sekumpulan perintah-perintah (*instruction*) yang diatur secara sistematis untuk menyelesaikan suatu masalah. Salah satu bahasa pemrograman komputer adalah bahasa pemrograman Java. Bahasa pemrograman Java adalah bahasa pemrograman berorientasi objek atau *Object Oriented Programming (OOP)*.

Java bersifat netral, tidak terikat pada suatu platform, dan mengikuti prinsip WORA (*Write Once Run Anywhere*). Beberapa bahasa pemrograman seperti Basic, Fortran, Pascal, COBOL, C, ProLog dan yang lainnya harus bergantung pada suatu platform karena mengharuskan Anda mengkompilasi program tersebut pada setiap sistem operasi yang ditentukan jika ingin menggunakan hasil program tersebut...”(Siallagan, Sariangin, 2009:13).

## II.2.1 Sifat-sifat Pemrograman Java

Sifat-sifat pemrograman Java, antara lain: berorientasi objek, terdistribusi, multiplatform, dan multithread.

### a. Java Berorientasi Objek

Pemrograman Java merupakan pemrograman yang berorientasi objek. Artinya, penulisan program harus dibuat dalam bentuk objek-objek, kemudian memodelkan sifat masing-masing objek dalam program. Java menentukan dan mengatur interaksi antara objek.

### b. Java Bersifat Terdistribusi

Program java sesuai dengan perkembangan teknologi saat ini, yaitu teknologi jaringan lokal seperti jaringan internet. Dalam hal ini, Java dapat digunakan untuk menyebarkan/mendistribusikan data atau informasi dalam workstation client, e-mail server, dan webserver (dalam jaringan komputer lokal dan internet).

### c. Java Bersifat Multiplatform

Program Java yang sudah diterjemahkan (dikompilasi) dengan benar dan berhasil dapat dijalankan pada platform yang lain atau pada sistem operasi yang lain. Java menggunakan prinsip WORA (Write Once Run Anywhere). Sekali menuliskan program maka hasil programnya dapat dijalankan pada sistem operasi yang mana saja. Artinya, sekali menulis program dalam bahasa Java dan mengkompilasinya maka program tersebut dapat dijalankan pada berbagai sistem operasi, seperti DOS, windows, dan Linux.

### d. Java Bersifat Multithread

*Thread* merupakan unit dasar penggunaan *Central Processing Unit* (CPU) yang terdiri dari *Thread\_ID*, program counter, register set, dan stack. *Multithread* adalah banyaknya pekerjaan yang dikerjakan dalam satuan waktu yang hampir bersamaan. Sebagai contoh, suatu web server mempunyai ratusan *client server* -beberapa client mengakses web server

secara bersamaan. Jika web server hanya mempunyai satu thread, berarti client yang lain harus menunggu. Oleh karena itu diperlukan multithread untuk melayani client server.

## II.2.2 Pemrograman J2ME

J2ME atau Java 2 Micro Edition merupakan kategori Java yang dirancang untuk menjalankan program Java pada perangkat seperti PDA atau handphone, yang memiliki karakteristik yang berbeda dengan computer biasa, misalnya memiliki memori yang terbatas, input yang terbatas, serta layar yang kecil.

J2ME sendiri merupakan kombinasi yang terbentuk antara sekumpulan interface Java yang sering disebut dengan Java API (Application Programming Interface) dengan JVM (Java Virtual Machine) yang didesain khusus untuk alat, yaitu JVM dengan ruang terbatas. Arsitektur J2ME mendefinisikan elemen-elemen untuk membentuk lingkungan Java runtime yang lengkap yang terdiri dari:

1. Lapisan Konfigurasi:

Configuration sendiri ditentukan perkembangannya oleh JCP (Java Community Process), inilah badan non-profit yang berkecukupan dengan perkembangan teknologi Java. J2ME mempunyai dua konfigurasi yaitu Connected Limited PhoneConfiguration (CLDC) dan Connected PhoneConfiguration (CDC). Pada bagian ini secara detail CLDC diperlukan untuk pengembangan aplikasi wireless dengan MIDP implementasinya CLDC digunakan untuk program Java pada perangkat keras dengan ukuran memori yang terbatas, pada 160 sampai dengan 512 Kilobyte.

2. Lapisan Profil:

Profile merupakan kebalikan dari configuration yaitu mengatur hal-hal yang spesifik untuk sebuah phone atau tipe market. Misalkan pada profile ini diatur tentang persistent storage dan UI. Saat ini JCP telah

mendefinisikan lima buah profile, salah satunya yaitu MIDP yaitu profile yang digunakan pada banyak mobile devices. Akibatnya, fitur-fitur yang kurang penting untuk diimplementasikan dalam handheld phone yang bersangkutan dari Java 2 harus dibuang. MIDP dikhususkan untuk digunakan pada handset dengan kemampuan CPU, memori, keyboard dan layar yang terbatas, seperti handphone, pager, PDA dan sebagainya. Aplikasi yang berjalan pada sebuah perangkat yang mendukung MIDP disebut dengan MIDlets, atau lebih singkatnya MIDlet merupakan aplikasi yang dibuat menggunakan Java 2 Micro Edition dengan profile Mobile Information PhoneProfile (MIDP).

#### **II.2.2.1 Connected Limited Phone Configuration (CLDC)**

CLDC atau Connected Limited Phone Configuration adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa library atau API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon selular, pager, dan PDA. Perangkat tsb dibatasi dengan keterbatasan memory, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal pada package, class, dan sebagian fungsi Java Virtual Machine yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat tersebut, JVM yang digunakan adalah KVM (Kilobyte Virtual Machine).

#### **II.2.2.2 CDC (Connected Device Configuration)**

*CDC (Connected Device Configuration)* merupakan komunitas proses pada Java yang memiliki standardisasi. CDC terdiri dari virtual machine dan library dasar untuk dipergunakan pada profile industri. Implementasi CDC pada J2ME adalah source code yang menyediakan sambungan dengan macam-macam platform.

#### **II.2.2.3 Mobile Information Phone Profile (MIDP)**

MIDP atau Mobile Information PhoneProfile adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan diatas CLDC, API

tambahan untuk daur hidup aplikasi, antar muka, jaringan, dan penyimpanan persisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan pada MIDP 2.0 dibandingkan MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan tone, tone sequence, dan file WAV walaupun tanpa adanya Mobile Media API (MMAPI).

#### **II.2.2.4 MIDlet**

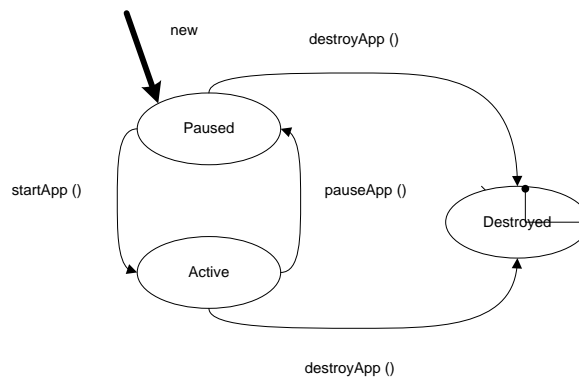
MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari class `javax. Microedition.midlet.MIDlet` yang didefinisikan pada MIDP. MIDlet berupa sebuah class abstrak yang merupakan subclass dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat bisa terbentuk

##### **II.2.2.4.1 Siklus MIDlet**

kehidupan MIDlet dimulai ketika di-instantiate oleh AMS. MIDlet pada awalnya masuk status “Pause” setelah perintah baru dibuat. AMS memanggil constructor public tanpa argumen dari MIDlet. Jika sebuah exception terjadi dalam constructor, MIDlet memasuki status “Destroyed” dan membuangnya segera. MIDlet masuk ke dalam status “Active” atas pemanggilan method `startUp()` oleh AMS. MIDlet masuk ke dalam status “Destroyed” ketika AMS memanggil method `destroyApp()`. Status ini juga kembali diakses ketika method `notifyDestroyed()` kembali dengan sukses kepada aplikasi. Dengan catatan bahwa MIDlet hanya bisa memasuki status “Destroyed” sekali dalam masa hidupnya.<sup>1</sup>

---

<sup>1</sup> JENI-Pengembangan Aplikasi Mobile



**Gambar 2: Daur hidup MIDlet**

### II.2.2.5 Fitur Ponsel Yang Mendukung J2ME

Fitur *mobile phone* yang mendukung J2ME adalah *mobile phone* yang mendukung konfigurasi dan profil dari J2ME. konfigurasi dan Profil adalah spesifik kemampuan dari paket perangkat yang akan anda jadikan target aplikasi anda, sebagai contoh saya punya HP K608 itu mempunyai spesifikasi CLDC 1.1 dan MIDP 2.0 artinya HP saya bisa menjalankan program dengan paket yang ada di CLDC 1.1 dan MIDP 2.0.

MIDP 1.0 seperti nokia 6015 dan nokia yang dengan slidenya yang saya lupa apa mereknya dan HP lama lainnya. MIDP 2.0 HP baru seperti nokia N70, N73, SE K600, SEW800 dan lain lain. Karena MIDP 2.0 lebih banyak ada baiknya anda pilih MIDP 2.0 dan CLDC 1.1.<sup>2</sup>

## II.3 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana

<sup>2</sup> <file:///G:/referensi%20TA%201/TA/88935-tutorial-membuat-program-java-j2me-di-hp.htm>

aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax* /semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch [1], metodologi coad [2], metodologi OOSE [3], metodologi OMT [4], metodologi shlaer-mellor [5], metodologi wirfs-brock [6], dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – <http://www.omg.org>). Tahun 1997 UML versi

1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999 [7] [8] [9]. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.<sup>3</sup>

### II.3.1 Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam tabel 2.

**Tabel 2. Konsep Dasar UML**

<b>Major Area</b>	<b>View</b>	<b>Diagrams</b>	<b>Main Concepts</b>
Structural	static view	class diagram	class, association, generalization, dependency,
	use case view	Use case diagram	use case, actor, association, extend, include, use case generalization
	implementati on view	component diagram	component, interface, dependency, realization

---

<sup>3</sup> <http://harmiprasetyo.wordpress.com/pengantar-uniified-modelling-language-uml>

Dynamic	state machine view	statechart diagram	state, event, transition,
	activity view	activity diagram	state, activity, completion
	interaction view	sequence diagram	interaction, object, message, activation
	colaborating diagram	collaborating, interaction, collaboration rule, message	model management
model management view	class diagram	package, subsystem, model	extensibility
All	All	constraint, stereotype, tagged values	

Untuk menguasai UML, sebenarnya cukup dua hal yang harus diperhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML.

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

- *use case diagram*
- *component diagram*
- *class analysis*

- *sequence diagram*
- *class diagram*

### II.3.2 Use Case Diagram

*Use case diagram* digunakan untuk memodelkan [bisnis](#) proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan [orang](#) yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk [elips](#) dengan [nama operasi](#) dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *Use Case*.<sup>4</sup> Untuk lebih lengkapnya, akan dijelaskan sebagai berikut.

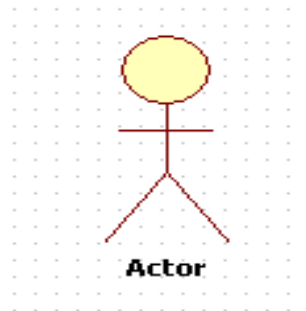
#### 1. *Actor*

*Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi masukan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.

**Contoh *actor* dapat dilihat pada gambar 3.**

---

<sup>4</sup> [http://id.wikipedia.org/wiki/UML#Use\\_Case\\_Diagram](http://id.wikipedia.org/wiki/UML#Use_Case_Diagram)



**Gambar 3. Actor**

## **2. Use Case**

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem memahami dan mengerti mengenai kegunaan sistem yang akan dibangun.

Cara menentukan *use case* dalam suatu sistem:

1. Pola perilaku perangkat lunak aplikasi.
2. Gambaran tugas dari sebuah *actor*.
3. Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.
4. Apa yang dikerjakan oleh suatu perangkat lunak (\*bukan bagaimana cara mengerjakannya).

Ada beberapa relasi yang terdapat pada *use case diagram*:

1. *Association*, menghubungkan *link* antar-elemen.
2. *Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

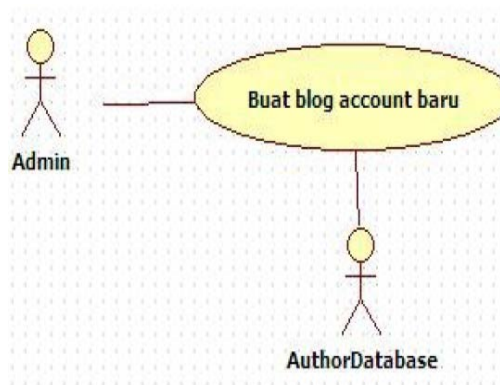
**Contoh *use case* dapat dilihat pada gambar 4.**



Gambar 4. Use Case

### 3. Contoh Use Case Diagram

Contoh *Use Case Diagram* dapat dilihat pada gambar 5.



Gambar 5. Use Case Diagram

### II.3.3 Kelas Analisis

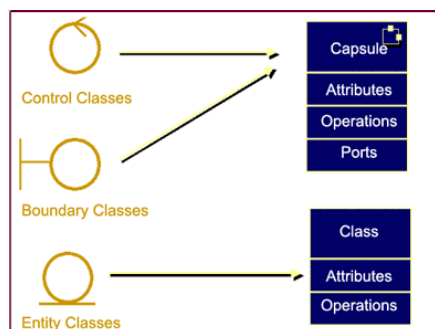
Kelas analisis di buat berdasarkan objek-objek yang terdapat pada skenario *use case* yang telah di buat. Kelas analisis ini terdiri dari tiga jenis objek, yaitu:

1. Kelas *Boundary* adalah kelas yang memodelkan interaksi antara satu atau lebih *actor* dengan sistem. Kelas *Boundary* memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas dengan dunia luar. Kelas *Boundary* dapat berupa:

1. *User interface* yang merupakan sarana komunikasi antara sistem dengan *user*, misalnya jendela (*window*) dalam GUI.
2. *System Interface* yang merupakan sarana komunikasi antara sistem dengan sistem informasi lainnya misalnya *communication protocol*.
3. *Phoneinterface* yang merupakan sarana komunikasi antara sistem dengan *phone*(alat), seperti printer, sensor dan sebagainya.

2. Kelas *Control* digunakan untuk memodelkan perilaku mengatur, khusus untuk satu atau beberapa *use case* saja. *Control Object* (*instance* dari kelas *control*) biasanya mengontrol objek lain. Perilaku sebuah *control object* erat hubungannya dengan realisasi suatu *use case* tertentu. *Control object* menjalankan realisasi dari *use case* tersebut. Tidak semua *use case* memerlukan *control object*.

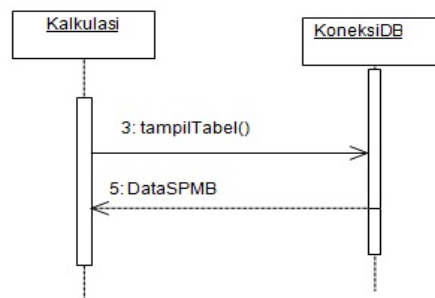
3. Kelas Entitas memodelkan informasi yang harus disimpan pada sistem. Kelas entitas memperlihatkan struktur data dari sebuah sistem. Oleh karena itu, kelas ini membantu kita untuk memahami apa yang kira-kira akan ditawarkan oleh sistem kepada *user*. Entitas objek (*instance* dari kelas entitas) biasanya bersifat pasif dan tetap. Tanggung jawab utama objek ini adalah untuk menyimpan dan mengatur informasi dalam sistem. Untuk contoh kelas analisis dapat dilihat pada gambar 6.



**Gambar 6. Kelas Analisis**

### II.3.4 Diagram Sekuen

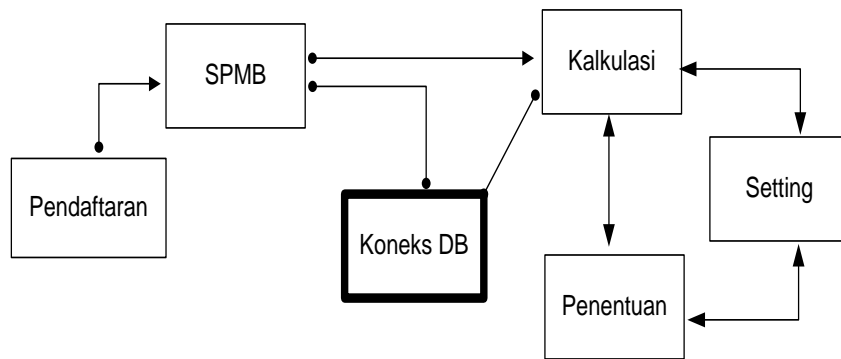
Diagram sekuen menggambarkan komunikasi/interaksi antar objek. Diagram ini menunjukkan sejumlah objek dan pesan (*message*) yang diletakkan diantara objek-objek didalam *use case* dan digambarkan melalui simbol-simbol. Garis panah menggambarkan simbol dari kegiatan yang dilakukan oleh objek atau menggambarkan komunikasi antar objek. Sedangkan *Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline*. *Activation* mengindikasikan sebuah objek yang akan melakukan sebuah aksi. *Lifeline* mengindikasikan keberadaan sebuah objek. Notasi untuk *Lifeline* adalah garis putus-putus vertikal yang ditarik dari sebuah objek. Untuk contoh diagram sekuen dapat dilihat pada gambar 7.



Gambar 7. Diagram Sekuen

### II.3.5 Diagram Kelas

Diagram Kelas merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Diagram Kelas menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Untuk contoh diagram kelas dapat dilihat pada gambar 8.




**Gambar 8. Diagram Kelas**

Hubungan antara kelas terdiri dari empat kardinalitas, untuk lebih jelasnya dapat dilihat pada Tabel 3.

**Tabel 3. Kardinalitas**

<b>Nama</b>	<b>Lam</b>	<b>Keterangan</b>
One to one	↔	Kelas pertama hanya dapat mengakses satu objek pada kelas satunya, begitu juga sebaliknya.
One to many	←•	Kelas pertama hanya dapat mengakses satu objek pada kelas satunya, namun kelas satunya dapat mengakses banyak objek pada kelas pertama.
Many to one	•→	Kelas pertama dapat mengakses banyak objek pada kelas satunya, namun kelas satunya hanya dapat mengakses satu objek pada kelas pertama.

Many to many		Kelas pertama dapat mengakses banyak objek pada kelas satunya, begitu juga sebaliknya.
--------------	---	--

## II.4 NetBeans

NetBeans mengacu pada dua hal, *platform* untuk pengembangan aplikasi *desktop* Java dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan *platform* NetBeans.

*Platform* NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak *moduler* yang disebut modul. Modul dapat dikembangkan secara independen, aplikasi berbasis *platform* NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerfull*.

### II.4.1 Platform NetBeans

*Platform* NetBeans adalah *framework* yang dapat digunakan kembali untuk menyederhanakan pengembangan aplikasi *desktop*. Ketika aplikasi berbasis *platform* NetBeans dijalankan, kelas `Main` dari *platform* akan dieksekusi. Modul-modul yang tersedia ditempatkan di dalam memori dan tugas *startup* modul dijalankan.

Aplikasi dapat menginstal modul secara dinamis dan juga dapat memasukkan modul *Update Center* untuk mengijinkan pengguna aplikasi men-*download digitally-signed upgrade* dan fitur-fitur baru secara langsung ke dalam aplikasi yang berjalan. Penginstalan kembali sebuah *upgrade* atau rilis baru tidak memaksa pengguna untuk men-*download* keseluruhan aplikasi lagi.

Fitur-fitur yang disediakan oleh *platform* NetBeans:

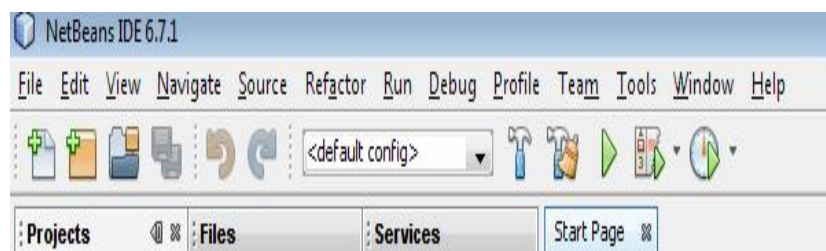
- a. Manajemen antarmuka (misal: menu & *toolbar*).
- b. Manajemen pengaturan pengguna.
- c. Manajemen penyimpanan (menyimpan dan membuka Berbagai macam data).
- d. Manajemen jendela.
- e. *Wizard framework* (mendukung dialog langkah demi langkah).

#### II.4.2 NetBeans IDE

NetBeans IDE adalah IDE *open source* yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan *platform* NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, *Web*, EJB dan aplikasi *mobile*).<sup>5</sup>

Langkah-langkah untuk menjalankan NetBeans IDE, sebagai berikut:

1. Buka aplikasi NetBeans IDE (*Intergrated Development Environment*) 6.7.1
2. Pada IDE, pilihlah *file* > *New Project* (Ctrl-Shift-N)

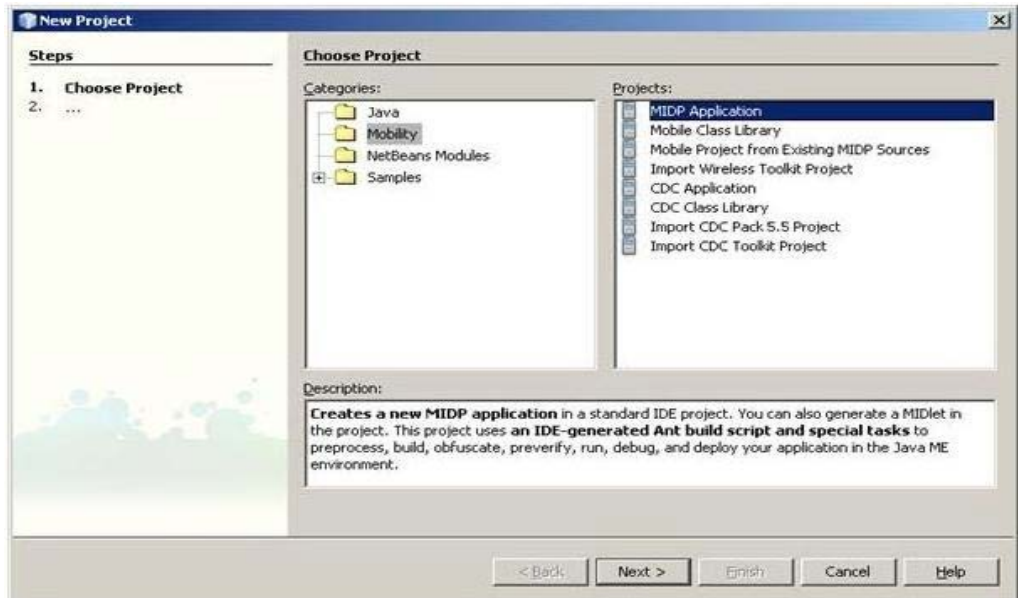


**Gambar 9. Toolbar pada NetBeans**

3. Pada *New Project* , pilih kategori *Mobility* dan dilanjutkan dengan meng-klik *next*

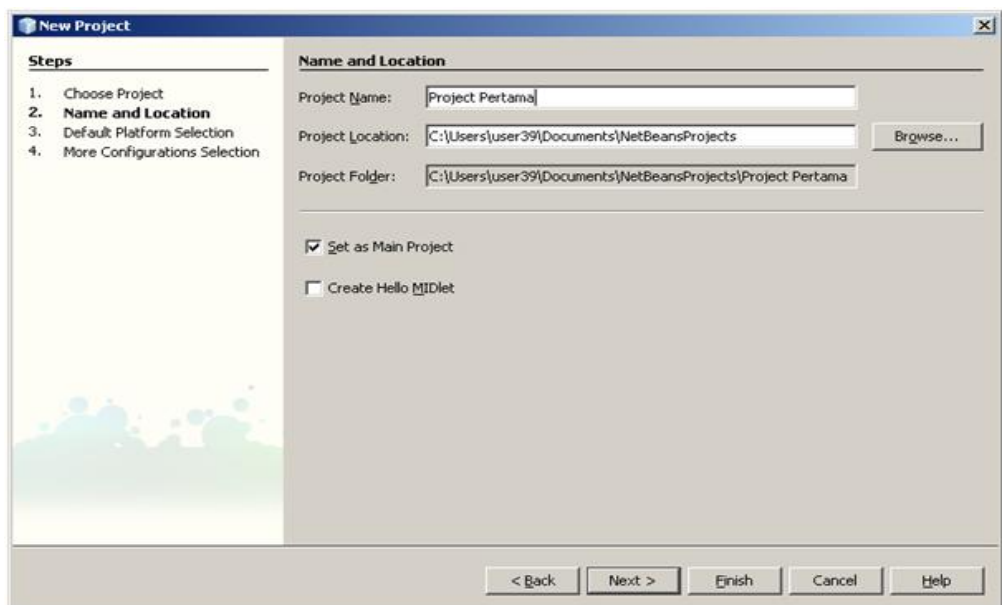
---

<sup>5</sup> <http://javastudy.wordpress.com>



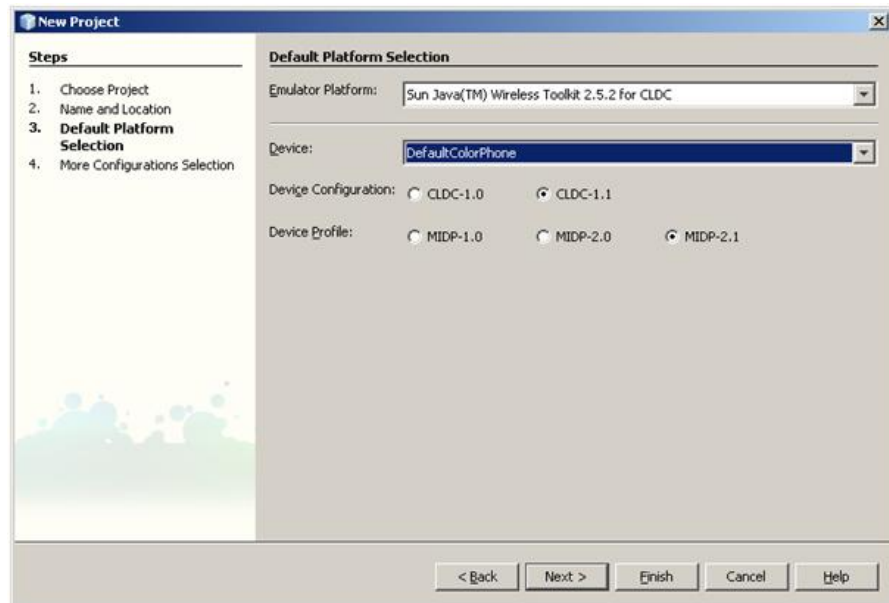
Gambar 10. Tampilan Choose Project

4. Pilih MIDP Application dan tekan **Next**. Selanjutnya anda akan di bawa ke layer Name and Location seperti di bawah.



Gambar 11. Tampilan Name and Location

5. Input Project Name yang sesuai dengan nama project anda dan hilangkan centang di Create Hello Midlet. Tekan **Next**



**Gambar 12. Tampilan Default Platform Selection**

Ini adalah layar untuk menentukan target perangkat yang akan anda buat aplikasinya. Pada baris Emulator Platform adalah pilihan emulator yang anda Install di computer anda, untuk saat ini anda hanya menginstall emulator bawaan dari NetBeans anda. Setelah anda merasa perlu menginstall emulator platform untuk spesifik dari vendor lain seperti Nokia, Sony Ericsson dan Motorola anda dapat mengubahnya disini.

6. Pilih Next dan langsung pilih Finish.

Kita mulai dengan membuat classMIDlet baru untuk project anda. Klik kanan para project anda di project explore dan pilih new-MIDlet, namakan class baru itu dengan nama MIDletPertama. Kode terakhir file MIDlet anda terlihat seperti dibawah ini.

```

    /*
     * To change this template, choose Tools | Templates
     * and open the template in the editor.
     */

    import javax.microedition.midlet.*;
    import javax.microedition.lcdui.*;

    /**
     * @author Puputs
     */
    public class MIDletPertama extends MIDlet {
        public void startApp() {

        }

        public void pauseApp() {

        }

        public void destroyApp(boolean unconditional) {

        }
    }

```

**Gambar 13. Source code pada J2ME**

Untuk pertama kali kita akan menampilkan “Hello J2ME World” pada layar HP. Ada dua jenis tampilan dalam MIDlet.

- Low Level menggunakan class Canvas
- High Level menggunakan Form

Disini kita akan menggunakan form untuk mempermudah anda mengerti pembentukan tampilan di MIDlet.

```

public void startApp() {
    Form frPertama = new Form("Judul");
    frPertama.append("Hello J2ME World");
    Display.getDisplay(this).setCurrent(frPertama);
}

```

**Gambar 14. Source code pada J2ME**

Perhatikan kode di atas, frPertama adalah object Form yang akan kita tampilkan di layer, kita memberi judulForm tersebut dengan “Judul”. Lalu method append adalah method untuk menambahkan Item pada Form, Item

yang ingin saya tambahkan adalah String “Hello J2ME World”. Statement Display adalah statement untuk menampilkan object frPertama kedalam layer.

7. Pilih F11 untuk **build** project dan F6 untuk **run project**. Pilih midlet yang ingin dijalankan dan tekan launch di emulator, selanjutnya emulator anda akan menampilkan Form yang anda buat, seperti di bawah.



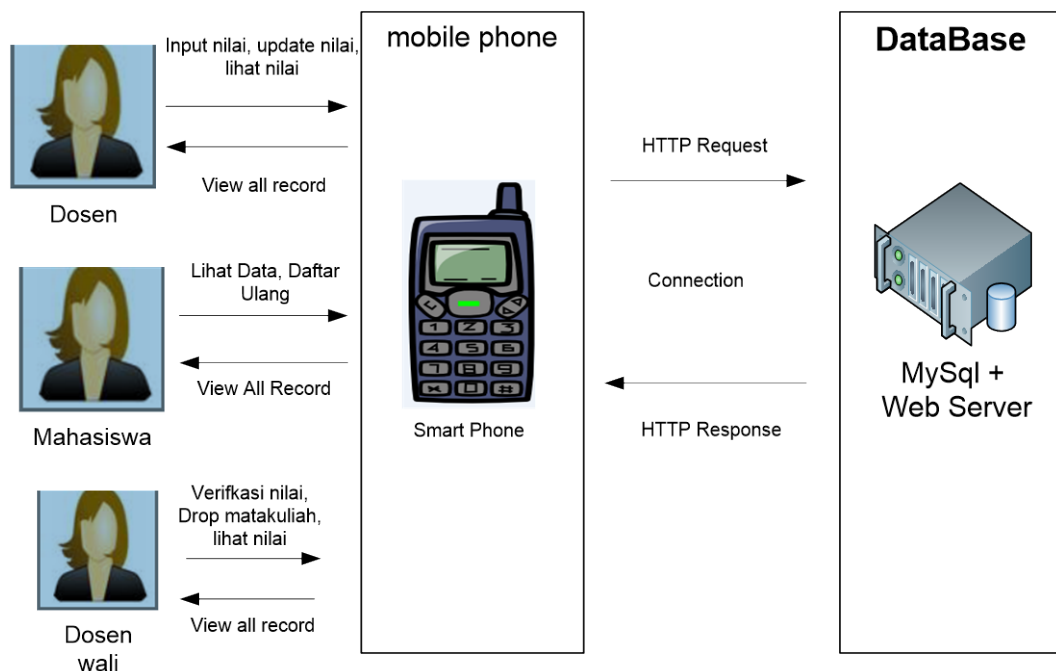
**Gambar 15. Tampilan Emulator**

### Bab III Analisis dan Perancangan

Bagian ini memuat uraian tentang langkah-langkah penyelesaian masalah. Judul bab disesuaikan dengan judul. Jika diperlukan, bagian ini dapat terdiri lebih dari satu bab. Berikan pengantar di sini

#### III.1 Diskripsi Umum Aplikasi m-Simpol

Deskripsi umum aplikasi m-Simpol ini menggambarkan interaksi antara pengguna dan sistem. Untuk lebih lanjutnya tentang deskripsi umum aplikasi dapat dilihat, pada Gambar III.1 Deskripsi umum sistem.



Gambar 16. Deskripsi Umum Sistem

Keterangan :

Pengguna sistem mobile phone di atas ada 3 pengguna yaitu dosen, dosen wali dan mahasiswa. Jika pengguna adalah dosen, yang dapat dilakukan pada sistem ini adalah dosen dapat menginput nilai, update nilai dapat melihat nilai menggunakan mobile phone. Jika pengguna dosen wali sistem dapat melakukan verifikasi, drop matakuliah, dan melihat nilai menggunakan mobile phone. Jika pengguna adalah

mahasiswa, sistem dapat melakukan daftar ulang dan melihat nilai menggunakan mobile phone.

### III.2 Kategori pengguna aplikasi

**Tabel 4.** Tingkatan pengguna aplikasi.

Jabatan	Aktifitas	Pengguna Aplikasi
Dosen	<p><b>Input nilai</b> : Dosen menginput nilai mahasiswa.</p> <p><b>Lihat nilai</b> : Dosen Melihat seluruh nilai mahasiswa berdasarkan matakuliah yang diajarkan.</p> <p><b>Edit nilai</b> : Dosen mengedit nilai mahasiswa yang salah.</p> <p><b>View all record</b>: Menampilkan data mahasiswa sesuai yang dipilih dosen.</p>	Admin
Dosen Wali	<p><b>Verifikasi</b> : Dosen wali mengverifikasi / menyetujui matakuliah yang diambil mahasiswa.</p> <p><b>Drop</b> : Dosen menghapus / mendedrop matakuliah yang tidak diverifikasi.</p> <p><b>Lihat nilai</b> : Dosen Melihat seluruh nilai mahasiswa berdasarkan matakuliah yang diajarkan.</p> <p><b>View all record</b>: Menampilkan data mahasiswa sesuai yang dipilih dosen.</p>	Admin
Mahasiswa	<p><b>Lihat nilai</b>: Mahasiswa melihat nilainya tiap-tiap semester yang sudah diambilnya.</p> <p><b>Daftar Ulang</b>: mahasiswa melakukan daftar ulang .</p> <p><b>View all record</b>: Menampilkan data mahasiswa sesuai yang dipilih mahasiswa.</p>	User

### **III.3 Fitur Utama Perangkat Lunak**

### **III.4 Kebutuhan Fungsional**

Kebutuhan Fungsional pada aplikasi yaitu :

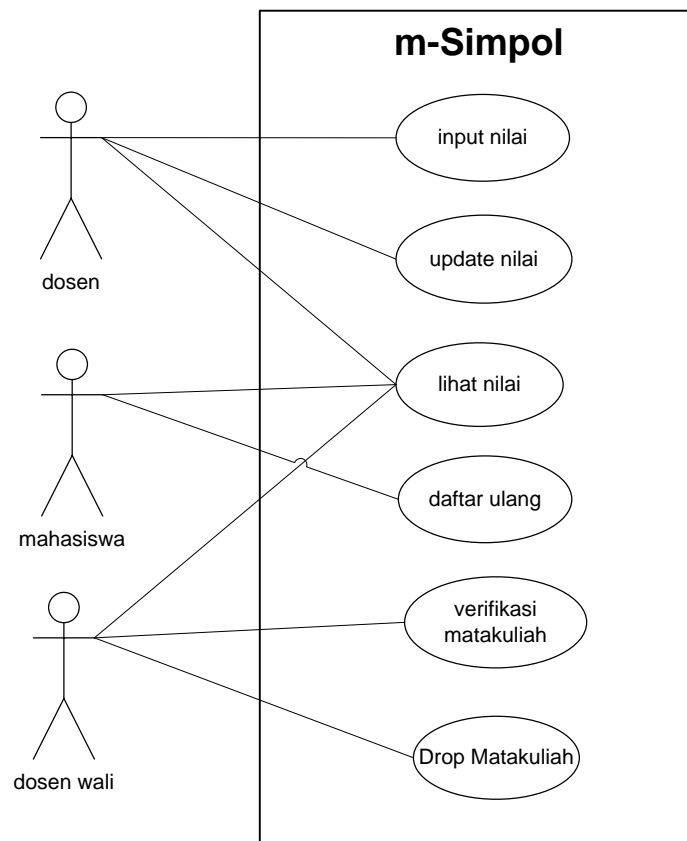
- F-001 Sistem mampu menyimpan semua data akademik ke dalam database.
- F-002 Sistem mampu menginput nilai menggunakan mobile device.
- F-003 Sistem mampu mengupdate nilai mahasiswa.
- F-004 Sistem mampu melihat nilai mahasiswa.
- F-005 Sistem mampu melakukan daftar ulang.
- F-006 Sistem mampu melakukan verifikasi daftar ulang.
- F-007 Sistem mampu membatalkan/drop matakuliah.

### **III.5 Batasan Sistem**

1. Sistem ini dijalankan menggunakan *wireless* atau pun *GPRS*.
2. Pada sistem ini bahasa pemrograman yang dipakai *J2ME*.
3. *Database* menggunakan *mysql*.

## III.6 Use Case

### III.6.1 Diagram Use Case



**Gambar 16. Diagram Use Case Aplikasi m-Simpol**

Keterangan : Pada diagram use case ini menjelaskan interaksi antar pengguna dengan aplikasi. Untuk lebih lanjutnya tentang interaksi pengguna per *use case* dapat dilihat pada skenario *use case*.

### III.6.2 Skenario Use Case

Pada skema diatas menjelaskan interaksi antara pengguna dengan sistem. Jika hak akses dosen, dosen akan Melakukan input data berupa nilai matakuliah yang diajar oleh dosen tersebut, nilai tersebut disimpan ke dalam database melalui *mobile device*. Jika nilai tersebut ada yang salah dosen akan melakukan periksa nilai untuk melihat nilai yang salah. Setelah itu dosen akan melakukan update nilai untuk merubah data yang salah tersebut. Jika hak akses adalah mahasiswa sistem akan menampilkan layar lihat nilai dan daftar ulang, jika mahasiswa

memilih lihat nilai sistem akan menampilkan nilai setiap matakuliah sesuai semester yang dipilihnya. Jika layar daftar ulang yang dipilih sistem akan menampilkan setiap matakuliah persemester, kemudian mahasiswa akan memilih matakuliah tersebut. Jika hak akses adalah dosen wali sistem dapat melakukan verifikasi, drop dan lihat nilai. Verifikasi berguna untuk menyetujui matakuliah yang diambil oleh mahasiswa, sedangkan untuk drop matakuliah berguna untuk menghapus matakuliah yang salah diambil oleh mahasiswa. Jika layar lihat nilai yang dipilih maka sistem akan menampilkan nilai-nilai mahasiswa yang ingin dilihat.

#### **III.6.2.1 Use case Input Nilai**

- Kondisi awal : *User* belum memilih matakuliah dan sistem belum menampilkan nama-nama mahasiswa.
- Kondisi akhir : Sistem menampilkan nama-nama mahasiswa yang mengambil matakuliah.
- Skenario : *User* memasukkan nilai mahasiswa dengan cara memilih A-E, kemudian sistem akan menyimpannya ke dalam *database*.

#### **III.6.2.2 Use case Update Nilai**

- Kondisi awal : Sistem belum mengubah nilai apa pun
- Kondisi akhir : Sistem mengubah data dan menyimpannya ke dalam *database*.
- Skenario : *User* memasukan NIM dan memilih matakuliah mahasiswa yang ingin diubah kemudian sistem mencari ke dalam *database* dan menampilkan ke layar nilai yang lama dan user mengubah nilai yang lama ke nilai yang baru kemudian sistem menyimpannya kembali ke dalam *database*.

### **III.6.2.3 Use case Lihat Nilai**

- Kondisi awal : *User* memasukkan semester berapa yang akan dilihat.
- Kondisi akhir : Sistem menampilkan nilai yang diminta oleh *user*.
- Skenario : *User* memasukan semester berapa yang akan akan lihat kemudian *user* akan mengklik tombol cari untuk mencari nilai mahasiswa. sistem menampilkan nilai-nilai mahasiswa per semester.

### **III.6.2.4 Use case Daftar ulang**

- Kondisi awal : *User* memasukan semester berapa yang akan didaftar ulang.
- Kondisi akhir : Sistem meyimpan matakuliah yang didaftar ulang ke dalam database.
- Skenario : *User* memilih semester berapa yang akan didaftar ulang, kemudian sistem menampilkan matakuliah yang akan didaftar ulang sesuai semester yang dipilih. Selanjutnya *user* akan memilih matakuliah yang akan diambilnya, kemudian sistem akan menyimpannya ke dalam *database*.

### **III.6.2.5 Use case Verifikasi**

- Kondisi awal : Sistem belum mengverifikasi matakuliah yang diambil mahasiswa.
- Kondisi akhir : Sistem mengverifikasi matakuliah mahasiswa kemudian disimpan ke dalam *database*.

Skenario : User memasukan nim mahasiswa dan memilih semester berapa yang akan diverifikasi, kemudian sistem akan menampilkan matakuliah yang akan diverifikasi, lalu user akan memilih matakuliah yang akan diverifikasi dan sistem akan menyimpannya ke dalam database

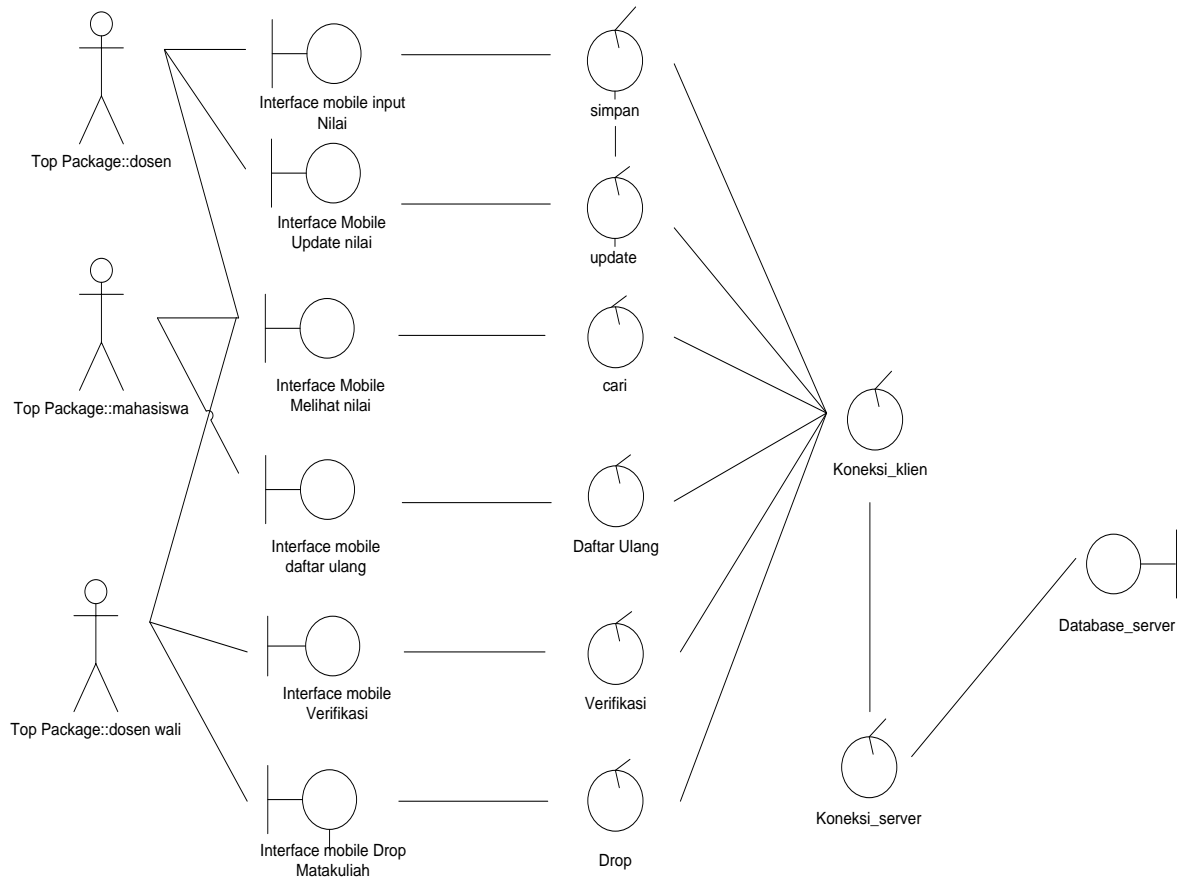
#### **III.6.2.6 Use case Drop Matakuliah**

Kondisi awal : *User* memasukkan nim dan semester.

Kondisi akhir : Sistem menghapus matakuliah.

Skenario : *User* memasukan nim mahasiswa dan semester, sistem menampilkan matakuliah yang telah diverifikasi. Kemudian *user* memilih matakuliah yang ingin dihapus, selanjutnya sistem akan menghapus matakuliah tersebut dan menyimpannya kembali ke dalam *database*.

### III.7 Analisis Kelas



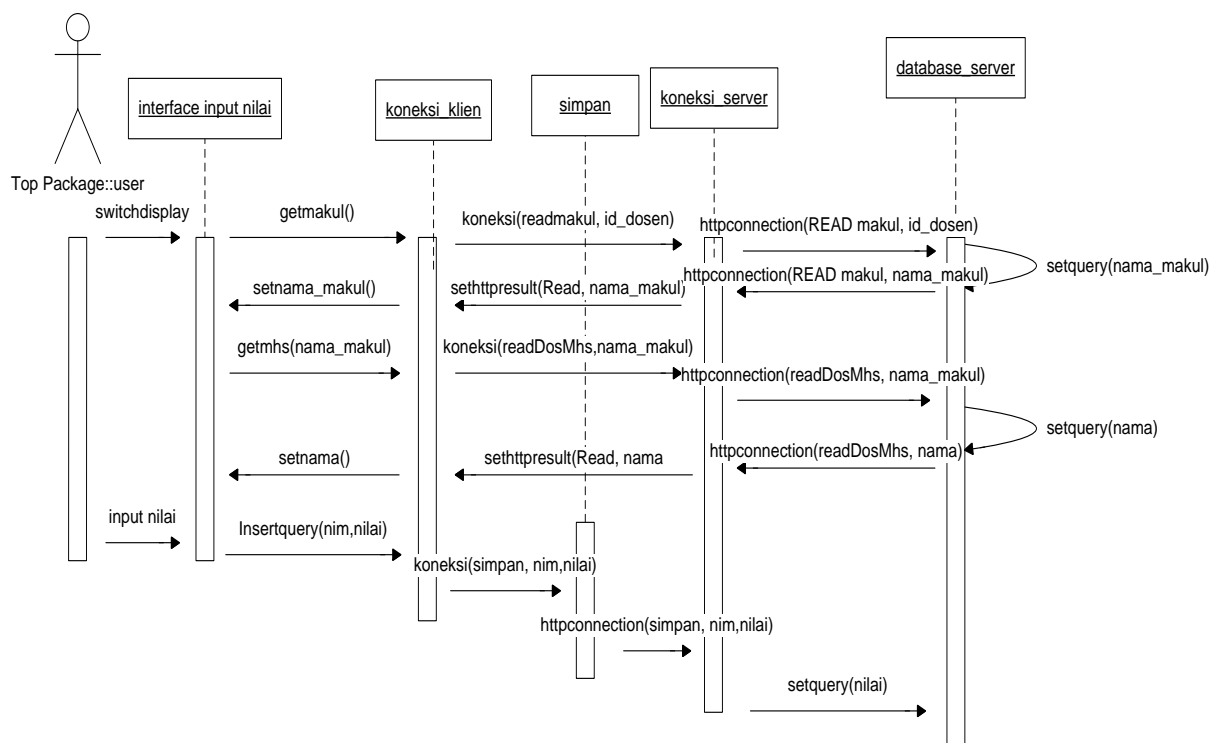
Gambar 16. Analisis Kelas

## Bab IV Perancangan

*Sequence Diagram* digunakan untuk menjelaskan lebih rinci jalannya aplikasi setiap *use case*. diagram ini merupakan gambaran dari skenario *use case* pada bab sebelumnya yang merupakan respon dari sebuah even untuk menghasilkan *output* tertentu.

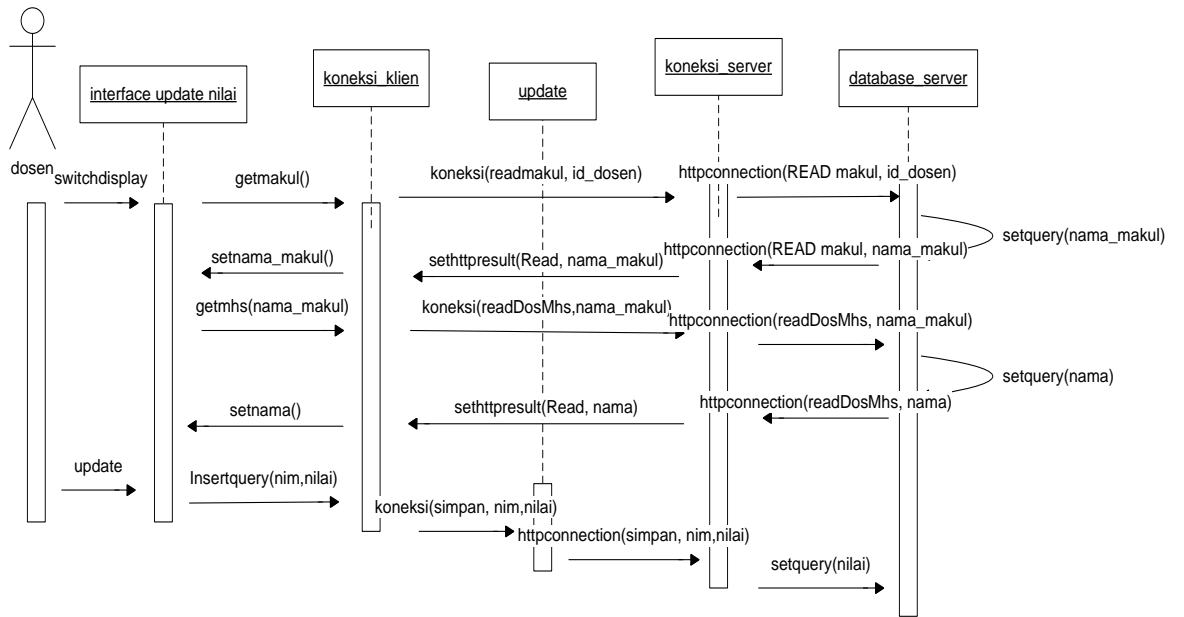
### IV.1 Sequence Diagram

#### IV.1.1.1 Sequence Diagram Use Case Input Nilai



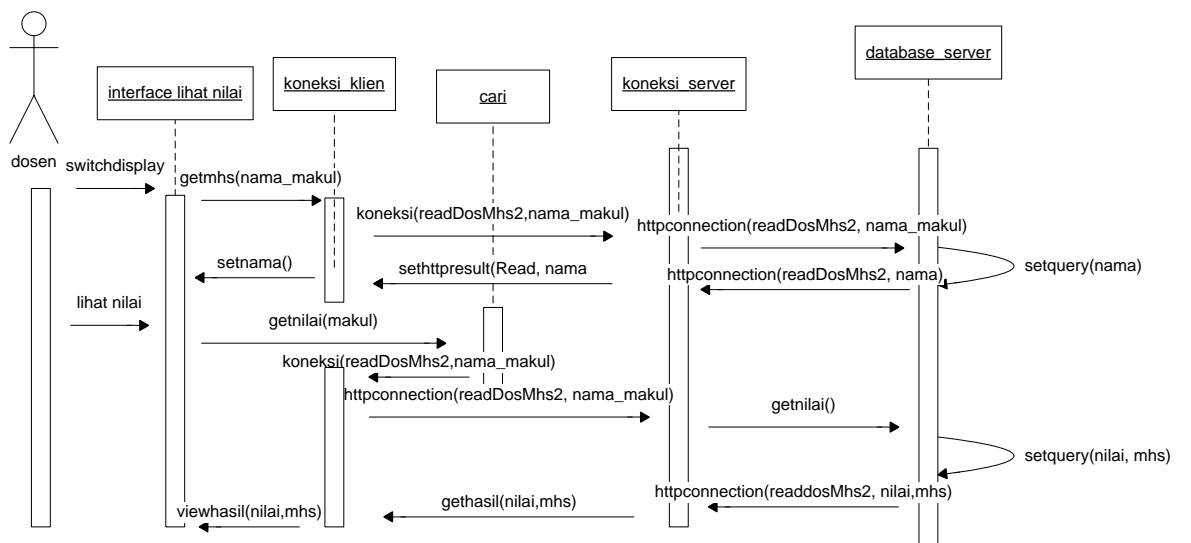
Gambar 17. Sequence Diagram Use Case Input Nilai

### IV.1.1.2 Sequence Diagram Use Case Update nilai



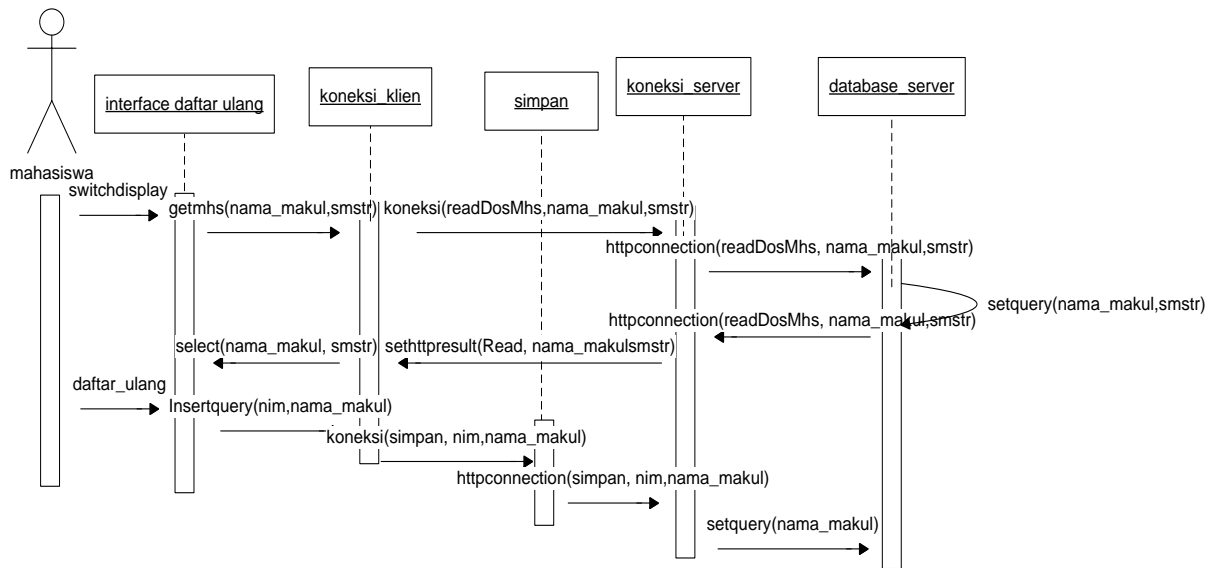
Gambar 18. Sequence Diagram Use Case Update Nilai

### IV.1.1.3 Sequence Diagram Use Case Lihat Nilai



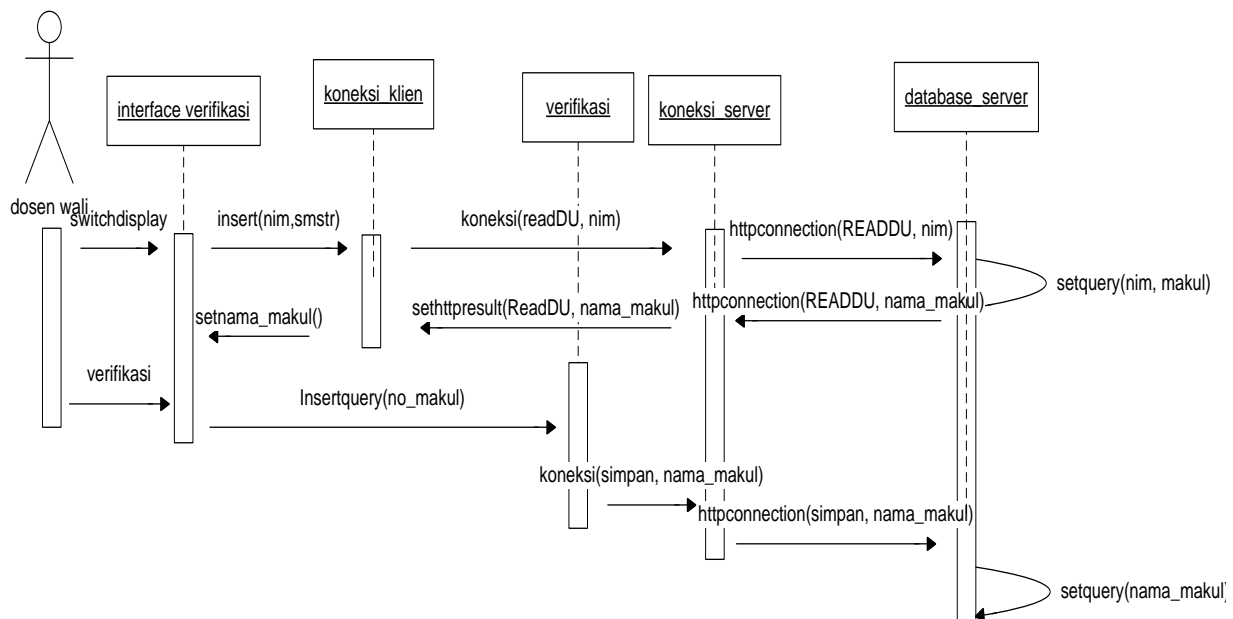
Gambar 19. Sequence Diagram Use Case Lihat Nilai

#### IV.1.1.4 Squence Diagram Use Case Daftar Ulang



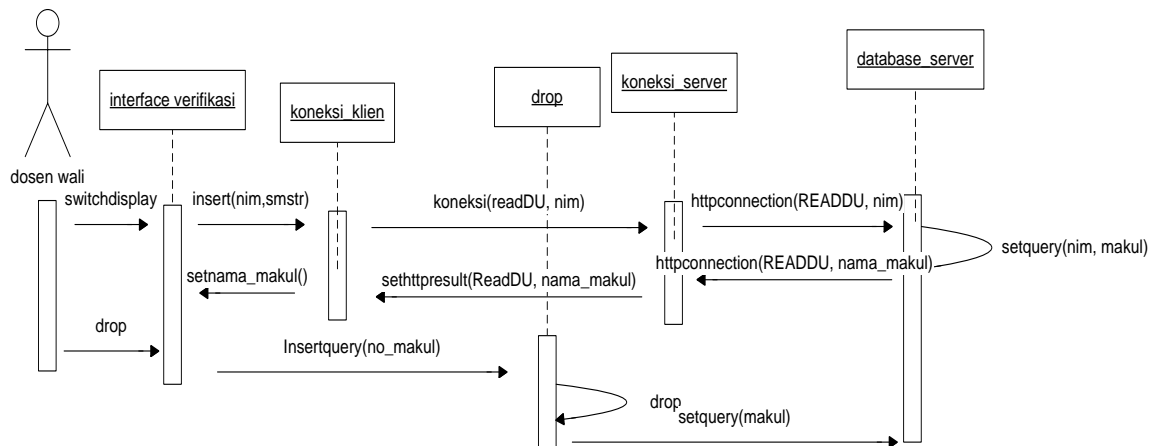
Gambar 20. Squence Diagram Use Case Daftar Ulang

#### IV.1.1.5 Squence Diagram Use Case Verifikasi Matakuliah



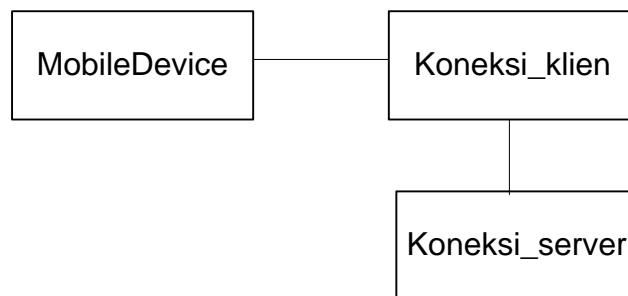
Gambar 21. Squence Diagram Use Case Verifikasi

#### IV.1.1.6 Squence Diagram Use Case Drop Matakuliah



Gambar 22. Squence Diagram Use Case Drop Matakuliah

#### IV.2 Diagram Kelas



Gambar 25. Diagram kelas

##### IV.2.1 Gambar Digram Kelas m-Simpol

Ket : Gambar 25 menjelaskan hubungan antar kelas dalam sistem yang akan dibuat dalam suatu diagram kelas. Mulai dari aliran data dari *boundary* kepada *controller* dan *controller* kepada *entity-entity* yang terkait dengan proses pada *controller* tersebut.

### IV.3 Rancangan Kelas Rinci

#### IV.3.1 Kelas Mobile device

<b><i>MobileDevice</i></b>
-
+ initialize() + startMIDlet() + resumeMIDlet() + switchDisplayable() +commandAction for Displayable() + exitMIDlet() +startApp() +pauseApp() +destroyApp() +setHttpResult() +koneksi() +setCurrentDisplay()

#### IV.3.2 Kelas Koneksi klien

<b><i>Koneksi_klien</i></b>
-
+run() +stop() +set +setHttpResult()

#### IV.3.3 Kelas Koneksi Server

<b><i>Koneksi_Server</i></b>
-
+DataNilai() +TampilData() +DataDU() +UpdateNilai() +deleteData() +cariMK() +cariMakulDos() +cariMakulDos2() +login()

## IV.4 Perancangan Antar Muka

### IV.4.1 Login dosen

#### IV.4.1.1 Rancangan Tampilan Login Dosen



Gambar 26. Tampilan Login Dosen

Gambar 26 Layar mobile device ini merupakan layar login dosen. Dosen memasukkan username dan password yang sesuai, jika berhasil layar mobile device akan menampilkan pesan “anda user sebagai dosen”.

#### IV.4.1.2 Algoritma/Query

Nama Kelas : Login.class

Nama Operasi : Login

Algoritma :

```
// input username & input password
// melakukan koneksi ke Http
Koneksi (“READLOGIN”, new string[] ( username & pass ))
// hasil pengembalian
Hasil = username, pass, hak_akses
// pembagian hak akses
If(hak_akses=dosen)
  Then goto menu dosen;
```

## IV.4.2 Login dosen wali

### IV.4.2.1 Rancangan Tampilan dosen wali



Gambar 27 Tampilan Login dosen wali

Gambar 27 Layar mobile device ini merupakan layar login dosen wali. Dosen wali memasukkan username dan password yang sesuai, jika berhasil layar mobile device akan menampilkan pesan “anda login sebagai dosen wali”.

### IV.4.2.2 Algoritma/Query

Nama Kelas : Login.class

Nama Operasi : Login

Algoritma :

```
// input username & input password
// melakukan koneksi ke Http
Koneksi (“READLOGIN”, new string[] ( username & pass ))
// hasil pengembalian
Hasil = username, pass, hak_akses
// pembagian hak akses
If(hak_akses=mhs)
    Then goto menu_dosen_wali;
```

### IV.4.3 Login Mahasiswa

#### IV.4.3.1 Rancangan Tampilan Login Mahasiswa



Gambar 27 Tampilan Login Mahasiswa

Gambar 27 Layar mobile device ini merupakan layar login mahasiswa. Mahasiswa memasukkan username dan password yang sesuai, jika berhasil layar mobile device akan menampilkan pesan “anda user sebagai mhs”.

#### IV.4.3.2 Algoritma/Query

Nama Kelas : Login.class

Nama Operasi : Login

Algoritma :

```
// input username & input password
// melakukan koneksi ke Http
Koneksi (“READLOGIN”, new string[] ( username & pass ))
// hasil pengembalian
Hasil = username, pass, hak_akses
// pembagian hak akses
If(hak_akses=mhs)
    Then goto menu mhs;
```

## IV.4.4 Input Nilai

### IV.4.4.1 Rancangan Tampilan Input Nilai



Gambar 28 Tampilan Input Nilai

#### IV.4.4.2 Algoritma/Query

Nama Kelas : Mobile Device.class

Nama Operasi : Cari mahasiswa dan save

Algoritma :

```
// tampil matakuliah berdasarkan matakuliah yang diijarkan oleh dosen
Koneksi ("READMAKULDOS", new string[]{dosen})
// kirim hasil ke setHttpresult
setHttpresult("READMAKULDOS", hasil)
// tampilkan hasil ke midlet
// pilih matakuliah
// cari ke database melalui http koneksi
Koneksi (READDOSMHS, new string[]{matakuliah})
Sethttpresult ("READDOSMHS", hasil)
// tampilkan hasil ke midlet
// masukkan nim dan nilai
// simpan melalui http request
Koneksi ("SIMPAN", new string[]{nim&nilai})
Sethttpresult ("SIMPAN", "tanda sukses")
```

## IV.4.5 Update Nilai

### IV.4.5.1 Rancangan Tampilan Update Nilai



Gambar 29. Tampilan Update Nilai

#### IV.4.5.2 Algoritma/Query

Nama Kelas : Mobile Device.class  
Nama Operasi : Cari nilai dan simpan  
Algoritma :

```
// tampil matakuliah berdasarkan matakuliah yang diijarkan oleh dosen
Koneksi ("READMAKULDOS", new string[] { dosen })
// kirim hasil ke setHttpresult
setHttpresult("READMAKULDOS", hasil)
// tampilkan hasil ke midlet
// pilih matakuliah
// cari ke database melalui http koneksi
Koneksi (READDOSMHS1, new string[] { matakuliah, nim })
Sethttpresult ("READDOSMHS1", hasil)
// tampilkan hasil ke midlet
// pilih matakuliah dan masukkan nim mahasiswa
// cari ke database
Koneksi (READDOSMHS1, new string[] { matakuliah, nim })
Sethttpresult("READDOSMHS1", hasil[] { nilai })
// Tampilkan hasil ke midlet
// Masukkan nilai baru
Koneksi ("UPDATE", new string[] { nilai })
Sethttpresult("simpan")
```

## IV.4.6 Lihat Nilai

### IV.4.6.1 Rancangan Tampilan Lihat Nilai



Gambar 29. Tampilan Lihat Nilai

### IV.4.6.2 Algoritma/Query

Nama Kelas : Mobile Device.class

Nama Operasi : cari

Algoritma :

```
// koneksi ke database  
Koneksi (READMHS, new string[]{nim,semester})  
Sethttpresult ("READMHS", hasil)  
// tampil hasil ke midlet berupa nilai dan matakuliah
```

## IV.4.7 Daftar Ulang

### IV.4.7.1 Rancangan Tampilan Daftar Ulang



Gambar 30. Tampilan Daftar Ulang

### IV.4.7.2 Algoritma/Query

Nama Kelas : Mobile Device.class

Nama Operasi : cari dan save

Algoritma :

```
// koneksi ke database
Koneksi (READMK, new string[]{nim,semester})
Sethttpresult ("READMK", hasil)
// tampil hasil ke midlet
// pilih matakuliah yang akan di daftar ulang
// simpan daftar ulang
Koneksi ("INSERTDU", new string[]{no_daftar})
Sethttpresult("INSERTDU", data telah disimpan)
```

## IV.4.8 Verifikasi Matakuliah

### IV.4.8.1 Rancangan Tampilan Verifikasi Matakuliah



Gambar 31. Tampilan Verifikasi Matakuliah

### IV.4.8.2 Algoritma/Query

Nama Kelas : Mobile Device.class

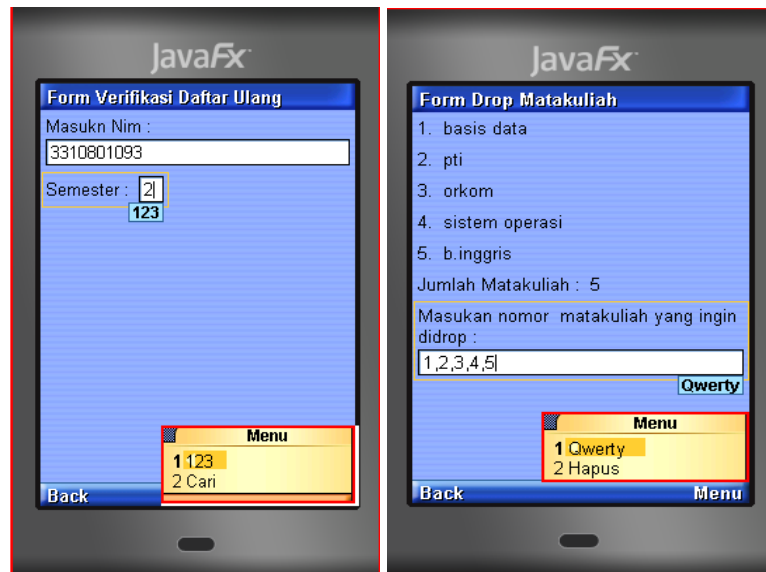
Nama Operasi : cari dan simpan

Algoritma :

```
Koneksi("sverifikasi", nim and semester)
//kirim data ke server
//server mencari data ke dalam database dan mengirimkan data ke midlet
Sethttpresult ("sverifikasi", hasil)
//tampilkan hasil ke layar verifikasi
Input no_makul
Koneksi ("Sverifikasi", no_makul)
//kirim data ke server
//server memproses verifikasi dan mengantar pesan error atau sukses
```

## IV.4.9 Drop MataKuliah

### IV.4.9.1 Rancangan Tampilan Drop Matakuliah



Gambar 32. Tampilan Drop Matakuliah

### IV.4.9.2 Algoritma/Query

Nama Kelas : Mobile Device.class

Nama Operasi : cari dan hapus

Algoritma :

```
Koneksi("sverifikasi", nim and semester)
//kirim data ke server
//server mencari data ke dalam database dan mengirimkan data ke midlet
Sethttpresult ("sverifikasi", hasil)
//tampilkan hasil ke layar verifikasi
Input no_makul
Koneksi ("delete", no_makul)
//kirim data ke server
//server memproses dan menghapus matakuliah
```

## Bab V Implementasi Dan Pengujian

Setelah dilakukan tahap perancangan maka tahap selanjutnya adalah implementasi dan pengujian pada perangkat lunak. Implementasi akan menghasilkan aplikasi yang dapat dijalankan di lingkungan operasional . Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai deskripsi perancangan maka perlu dilakukan pengujian. Pada perangkat keras akan dilakukan pengukuran dan pengujian untuk mengetahui kinerja sistem.

### V.1 Implementasi Kelas

Tabel 8 Implementasi kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	MobileDevice	MobileDevice.java	Mobile Device.jar
2	Connection	Connection.java	Connection.class
3	Function	Function.php	-

Ket : pada tabel di atas merupakan tabel implementasi kelas, memiliki 3 kelas yaitu kelas MobileDevice, Connection dan kelas function. Pada implementasi ini kelas-kelas yang digunakan sesuai dengan kelas perancangan yaitu memiliki 3 buah kelas. Pada sisi client memiliki 2 kelas yaitu MobileDevice dan Connection, untuk kelas MobileDevice pada perancangan kelas nya merupakan kelas midlet, sedangkan untuk connection merupakan kelas dari koneksi\_klien. Pada sisi web server nya terdapat kelas function, pada tahap perancangan kelas function merupakan kelas dari koneksi server.

### V.2 Implementasi Antar Muka

Tabel 9 Implementasi antar muka

No	Antar Muka	Nama File Fisik	Nama File Executable
1	Layar Mobile Device	MobileDevice.java	Mobile Device.jar

## V.3 Pengujian

### V.3.1 Identifikasi Pengujian

#### V.3.1.1 Pada tahap pengujian ini digunakan tiga metode pengujian.

##### 1. Pengujian dengan menggunakan satu laptop.

Untuk pengujian menggunakan satu laptop, hal-hal yang dibutuhkan laptop adalah sebagai berikut :

- *Emulator Netbeans* sebagai *platform* yang dapat menjalankan aplikasi m-Simpol.
- *DBMS MySQL* sebagai *database server* untuk penyimpanan data.
- *Apache* sebagai *web server*
- *URL = " http://localhost/taqu/function.php "*

##### 2. Pengujian dengan menggunakan dua laptop(*peer-to-peer*).

Pada pengujian ini, laptop dibagi menjadi dua bagian penggunaan yaitu laptop *server* dan laptop *client*.

2.1. Adapun hal-hal yang dibutuhkan pada laptop *server* adalah:

- *DBMS MySQL*
- *Wireless adhoc*
- *Setting IP*

A. Untuk di laptop server memiliki IP 192.168.3.24

2.2. Adapun hal-hal yang dibutuhkan pada laptop *client* adalah :

- *Emulator Netbeans*
- *Wireless adhoc*
- *Setting IP*

A. Untuk di laptop client kita setting IP nya menjadi 192.168.3.23

- *URL = " http://192.168.3.24/taqu/function.php "*

## **B. Pengujian dengan menggunakan *handphone*(GPRS).**

Dalam pengujian ini, dibutuhkan laptop *server* yang menjadi pusat seluruh layanan sistem akademik.

3.1. Adapun hal-hal yang dibutuhkan adalah :

- Fitur *CLDC*(*Connected Limited Device Configuratio*) sebagai konfigurasi untuk menjalankan program java.
- Fitur *MIDP*(*Mobile Information Device Profile*) sebagai penyedia librari java berupa jaringan, *interface*, *database* dan *timer*.
- *GPRS* atau *wireless* sebagai media untuk menghubungkan antara *handphone* dan *database server*.
- Proses hosting dilakukan pada file *database*, *function*, *config* dan *koneksi\_mysql*
- Nama domain = “*www. m.donnymesach.com*”

3.1.1 Pengujian pada *handphone* nokia X2

Tahap-tahap menjalankan m-SIMPOL pada nokia X2 adalah :

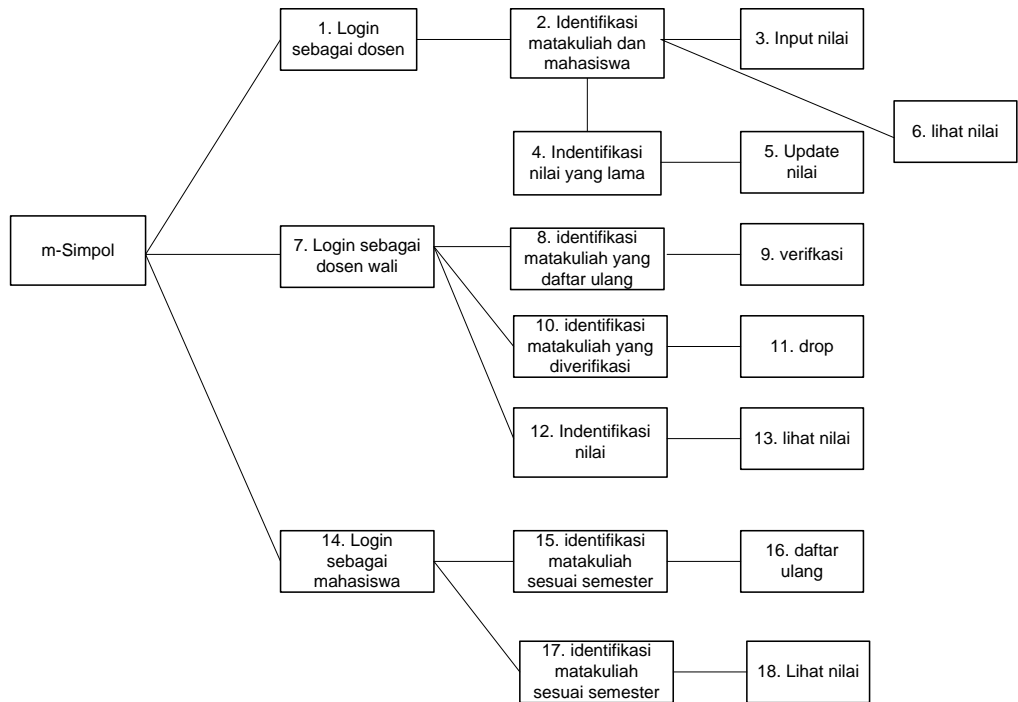
- *Setting GPRS* pada *handphone*.
- Pindahkan *file* SIMPOL.JAD ke *handphone*.
- m-SIMPOL dapat dijalankan.

3.1.2 Pengujian pada *handphone* blackberry 9300

Tahap-tahap menjalankan m-SIMPOL pada nokia X2 adalah :

- *Setting GPRS* pada *handphone*.
- Pindahkan *file* SIMPOL.JAD dari *netbeans* ke *handphone*.
- m-SIMPOL tidak dapat dijalankan, muncul pesan “*ErrorTunnel failed*”.

### V.3.2 Skenario Pengujian Aplikasi m-Simpol



Gambar 33 Skenario pengujian

## **Bab VI      Kesimpulan dan Saran**

### **VI.1 Kesimpulan**

Setelah di lakukan tahap analisis, perancangan, implementasi dan pengujian pada aplikasi m-Simpol, maka kesimpulan yang didapat adalah:

1. Aplikasi m-SIMPOL ini dapat diakses oleh pengguna sistem akademik diluar area Politeknik Negeri Batam menggunakan *GPRS*.
2. Aplikasi m-SIMPOL ini hanya dapat dijalankan di *Operating System handphone* tertentu yaitu *operating system symbian* dan *java*. Karena hasil pengujian dengan menggunakan *BlackBerry* tidak berhasil menjalankan aplikasi m-Simpol.

### **VI.2 Saran**

Adapun saran yang dapat diberikan untuk penyempurnaan dari aplikasi m-Simpol adalah sebagai berikut :

1. Pada penelitian selanjutnya diharapkan aplikasi m-SIMPOL dapat menangani keamanan jaringan.
2. Aplikasi dapat dijalankan pada berbagai jenis *mobile operating system*.

## Bab VII Daftar Pustaka

- [1] Kadir, Abdul. Bahasa Pemrograman Java™\_2. Yogyakarta: Andi Yogyakarta, 2004.
- [2] Raharjo, Budi., Heryanto, imam & Haryono, Arif. *Tuntunan Pemrograman Java untuk handphone*. Bandung: Informatika, 2007.
- [3] [http://id.wikipedia.org/wiki/UML#Use\\_Case\\_Diagram](http://id.wikipedia.org/wiki/UML#Use_Case_Diagram), diakses pada tanggal 23 Februari 2011.
- [4] Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak menggunakan UML dan Java*. Yogyakarta: Andi.
- [5] [http://id.wikipedia.org/wiki/Basis\\_data](http://id.wikipedia.org/wiki/Basis_data), diakses pada tanggal 25 Februari 2011.
- [6] <file:///G:/referensi%20TA%201/TA/JAVA%20e-Learning.htm>, diakses pada tanggal 1 Maret.
- [7] Pengantar Unified Modelling Language (UML) « Human Knowledge Belong To The World.htm, diakses pada tanggal 10 Maret 2011.