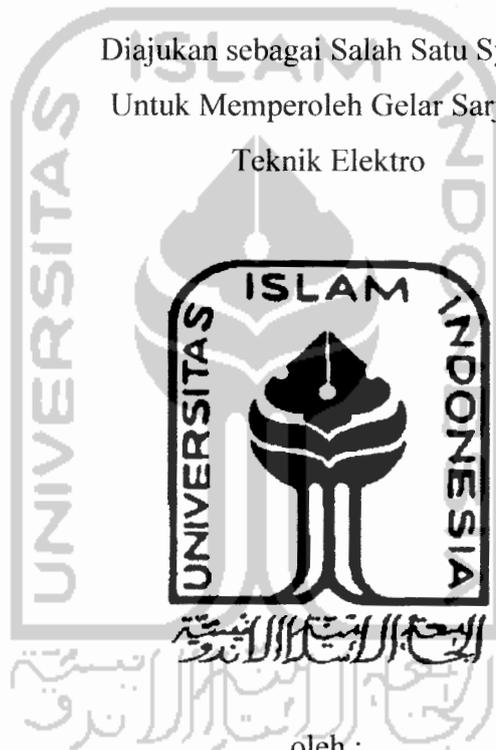


**PEMBUATAN APLIKASI MOBILE INFORMASI PINJAMAN
PERBANKAN SYARIAH DENGAN JAVA 2 MICRO EDITION**

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Teknik Elektro



oleh :

Nama : Angga Kusuma Bayu Aji

No. Mahasiswa : 03524024

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2007

LEMBAR PENGESAHAN PEMBIMBING

**PEMBUATAN APLIKASI MOBILE INFORMASI PINJAMAN
PERBANKAN SYARIAH DENGAN JAVA 2 MICRO EDITION**

TUGAS AKHIR



Pembimbing I,

(Tito Yuwono, ST., M.Sc)

Pembimbing II,

(Taufiq Hidayat, ST., M.CS)

LEMBAR PENGESAHAN PENGUJI

PEMBUATAN APLIKASI MOBILE INFORMASI PINJAMAN PERBANKAN SYARIAH DENGAN JAVA 2 MICRO EDITION

TUGAS AKHIR

oleh :

Nama : Angga Kusuma Bayu Aji

No. Mahasiswa : 03524024

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk

Memperoleh Gelar Sarjana Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia

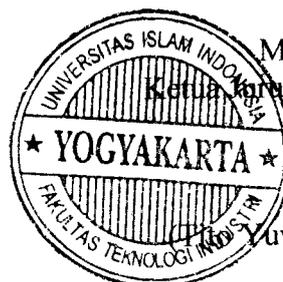
Yogyakarta, 03 Januari 2008

Tim Penguji,

Tito Yuwono, ST., M.Sc
Ketua

Taufiq Hidayat, ST., M.CS
Anggota I

Yusuf Aziz Amrulloh, ST
Anggota II



Mengetahui
Ketua Jurusan Teknik Elektro

(Tito Yuwono, ST., M.Sc)

HALAMAN PERSEMBAHAN

*Dengan penuh rasa syukur
Kupersembahkan Karyaku Ini Kepada*

*Allah SWT, Yang Maha Esa
Yang membuat semua hal sulit menjadi mudah, yang menjadikan
segala sesuatu menjadi mungkin, dan membuat rasa pedih menjadi
nikmat. Sujud syukurku aku panjatkan atas rahmat, hidayah dan
rezeki yang telah engkau berikan kepadaku*

*Ayahanda Edi Sutopo dan Ibunda Suhariyanti tercinta
Terima kasih untuk cinta, kasih sayang, pengorbanan, dukungan,
kesabaran, perhatian, kepercayaan, bimbingan, dan doa
yang tiada hentinya untukku*

*Adikku Siska yang selalu mengiringi dan mendukung hidupku
Yang selalu memberikan dorongan, cinta, dan kasih sayang
Terima kasih telah menjadi saudara yang selalu mengerti*

Special Thanks To

- © Allah SWT yang selama ini telah melimpahkan seluruh rahmat, hidayah dan ridhonya kepadaku
- © Papi Edi Sutopo dan Mami Suhariyanti tercinta terima kasih untuk semua cinta, kasih sayang, pengorbanan, dukungan, kesabaran, perhatian, kepercayaan, bimbingan, dan doa yang tiada hentinya untukku. Sehingga Angga bisa seperti sekarang ini. Semua yang angga lakukan hanya untuk membuat papi dan mami bahagia... I LOVE YOU...
- © Adikku siska. My Little SISTA yang paling aku sayangi terima kasih udah mau jadi satu-satunya saudara dalam hidupku. Semoga kita tetap akur terus and gak berantem lagi..oh iya kuliahnya diseriusin, cos tinggal kamu yang masih kuliah..ayo buat bangga ayah dan ibu..
- © Buat saudara sepupuku yang masih kuliah di Jogja mas Adji dan Shinta..mas cepetan lulus masak dari dulu kuliah gak lulus?.. kok kalah ma adikmu yang satu ini ^_^ (He3x)..terus buat shinta jangan sering berantem ya ma siska, kalian kan seperti pinang dibelah dua tapi beda bapak dan ibu saja..Cayo ya buat kalian..
- © Untuk Sweet Aprodithe yang pernah memberikan warna dalam kisah hidupku..terima kasih udah mau jadi seseorang yang mengerti, memahami, dan memberikan kasih sayang kepada aku..Walaupun kita tidak bisa

bertahan dan hanya sebentar..aku tetap masih sayang kamu..Aku cuma ingin kamu bahagia ada atau tanpa diriku..maafkan sifat egoisku..

© *All my best friend's, BAYU, MAMAD, JAEBRIK, BAJURI, HAVID, CHEPY, DINDA, and DEWI. 4 tahun waktu yang lama untuk mengenal kalian, walaupun kita jarang kumpul bareng tapi aku merasa kalianlah yang terbaik. Kalianlah yang terus memberikan Spirit and menambah indahny warna hidupku. Terima kasih buat Tuhan telah mempertemukan aku dengan kalian. Kalian akan selalu ada dihatiku, thank you guy's....!!!. Ingat semboyan kita (Friend's4Ever).*

© *Buat anak-anak kos kasunartan, Andi, Tomo, TegoeH, Wahyu, Aris. Terima kasih atas dukungan yang diberikan, semoga kalian semua sukses..Ingat kita Pernah satu kos, jadi aku akan menyempatkan waktu buat datang ke kos kalian.*

© *All my Great friend's, Azhar, Deni, Silly, Kempong, FAuzi, Trimbil, Arie, Erni, Berthy, Yeyen and Jati..Kenal kalian aku sangat berterima kasih, sebab tanpa kalian mungkin aku gak tau tempat2 asik buat nongkrong di Jogja..yah walaupun kadang harus bantingan terus Sampai duit hampir bokek, tapi nongkrong tetap jalan terus..Oh ya, dah jarang ni kita gak karaokean lagi, ditunggu ya kabarnya..Thank All, You Are My Sweet2x Friend's..*

© *Paling khusus buat mbak Afia..terima kasih ya udah jadi mentorku buat ngajarin J2ME..maaf kalau aku sering ngerepotin..semoga kita dapat sukses di masa depan..*

© Semua teman - teman Elektro UII angkatan 2003 yang tidak bisa disebutkan satu persatu.. Terima kasih buat bantuan dan kerjasama yang terjalin selama aku kuliah..

© Semua orang yang telah mengenalku dan membantu aku selama ini, dari aku masih kecil sampai dewasa sekarang ini.. Tak ada kata yang bisa aku ucapkan TERIMA KASIH..



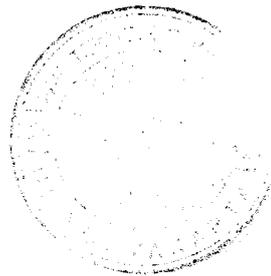
MOTTO

***Every one have a dream..not all dream
came true..cause dream can true, if
you try making true..not just
dreaming about it..***

***Tuhan menciptakan manusia berbeda -
beda dan tidak semua manusia itu
sempurna..hanya kesempurnaan itu
milik ALLAH SWT..***

***If you want something..Don't just
thinking..But do it..***

***Berbaktilah kepada orang
tuamu..Karena mereka telah
membesarkan kita dengan penuh kasih
sayang..***



KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr. Wb

Dengan mengucap rasa syukur Alhamdulillah kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga dapat terselesaikannya laporan Tugas Akhir (TA) yang berjudul Rancang Bangun Aplikasi Mobile Informasi Pinjaman Perbankan Syariah Dengan Java 2 Micro Edition (J2ME) dengan baik.

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik Elektro pada Universitas Islam Indonesia. Dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek. Sehingga ilmu yang didapat selama kuliah dapat diterapkan dalam kehidupan bermasyarakat dan dapat bermanfaat bagi bangsa Indonesia.

Penulis menyadari sepenuhnya dalam pengerjaan tugas akhir mendapat banyak bantuan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Tito Yuwono ST., M.Sc. selaku Ketua Jurusan Teknik Elektro dan selaku pembimbing I atas ketersediaan waktu membimbing penulis.
2. Bapak Taufiq Hidayat, ST., M.CS selaku dosen pembimbing II atas ketersediaan waktu membimbing penulis.

3. Seluruh Dosen pengajar Teknik Elektro UII, Ibu Budi, Ibu Ana, Bpk. Wahyudi, Bpk. Medila, Bpk. Yusuf, Bpk. Hendra, dan Bpk. Halim terima kasih atas segala ilmu pengetahuan yang diberikan selama kuliah di Teknik Elektro UII.
4. Teman – teman kampus yang tidak bisa disebut satu demi satu yang telah memberikan masukannya.

Semoga segala bantuan, dukungan dan bimbingan yang telah diberikan kepada penyusun mendapat imbalan dari Allah SWT.

Penulis menyadari dalam penulisan laporan tugas akhir ini masih jauh dari sempurna, karena keterbatasan kemampuan dan pengalaman. Penulis mengharapkan kritik dan saran yang bersifat membangun untuk memperbaiki tugas akhir ini. Semoga hasil dari tugas akhir ini dapat bermanfaat bagi semua pihak yang membutuhkan. Amin Ya Robbal 'Alamin.

Wassalamualaikum Wr. Wb.

Yogyakarta, Desember 2007

Penulis

DAFTAR ISI

Halaman Judul	i
Halaman Pengesahan Pembimbing	ii
Halaman Pengesahan Penguji	iii
Halaman Persembahan	iv
Halaman <i>Special Thank To</i>	v
Motto	viii
Kata Pengantar	ix
Daftar Isi	xi
Daftar Tabel	xiv
Daftar Gambar	xv
Abstrak	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Perancangan	2
1.5 Manfaat Perancangan	3
1.6 Sistematik Penulisan Laporan	3
BAB II LANDASAN TEORI	
2.1 Definisi Java	5

2.2	Pengertian J2ME	6
2.2.1	Bagian – Bagian Arsitektur J2ME	8
2.2.2	MIDlet	9
2.2.3	JAD (<i>Java Application Descriptor</i>)	13
2.3	Pengertian Bank	13
2.4	Pengertian Bank Syariah	14
2.4.1	Produk Pembiayaan Bank Syariah	14
2.4.2	Perhitungan Peminjaman.....	15
 BAB III PERANCANGAN SOFTWARE		
3.1	Metode Perancangan	19
3.2	Diagram Perancangan	19
3.3	Membuat dan Membuka Aplikasi	23
3.4	Setting Project	25
3.5	Hasil Perancangan	27
3.6	Perancangan Aplikasi	27
3.6.1	<i>Use Case Diagram</i>	28
3.6.2	<i>Class Diagram</i>	30
3.6.3	<i>Sequence Diagram</i>	31
3.7	Perancangan Antarmuka	33
3.7.1	Rancangan Menu Awal	33
3.7.2	Rancangan <i>Interface</i> Mudharabah	34
3.7.3	Rancangan <i>Interface</i> Murabahah	35
3.7.4	Rancangan <i>Interface</i> Petunjuk	36

BAB IV PENGUJIAN, ANALISA DAN PEMBAHASAN

4.1 Pengujian Aplikasi	37
4.1.1 Pengujian Umum Aplikasi	37
4.1.2 Tahapan Pengujian	38
4.2 Pengujian Sistem Melalui <i>Handphone</i>	39
4.2.1 Pengujian Pinjaman Mudharabah.....	39
4.2.2 Pengujian Pinjaman Murabahah.....	41
4.2.3 Pengujian Petunjuk.....	43
4.3 Analisa Pengujian	45
4.4 Hasil Analisa Menggunakan <i>Alfa Test</i>	45
4.5 Hasil Analisa Menggunakan Metode <i>Black Box Test</i>	46
4.5.1 Prosedur Normal Mudharabah & Murabahah	47
4.5.2 Prosedur Tidak Normal Mudharabah & Murabahah	47
4.5.3 Prosedur Normal Petunjuk	50
4.5.4 Prosedur Tidak Normal Petunjuk	50
4.6 Kelebihan dan Kekurangan Program	51

BAB V PENUTUP

5.1 Kesimpulan	53
5.2 Saran	53

DAFTAR PUSTAKA	55
-----------------------------	----

LAMPIRAN

DAFTAR TABEL

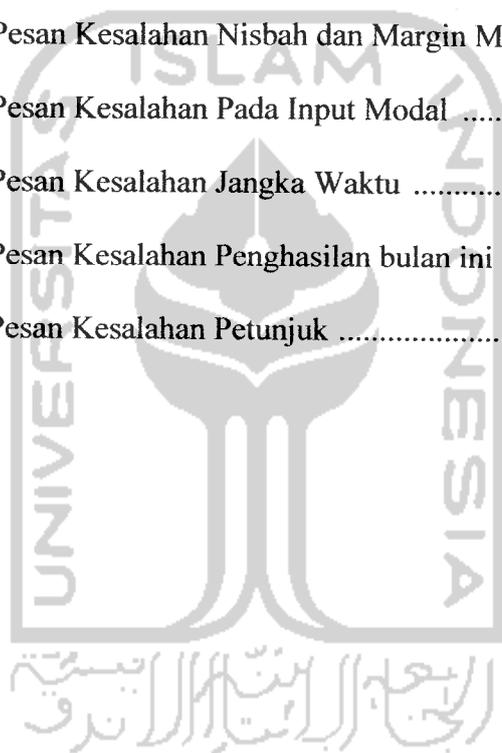
Tabel 2.1	Atribut <i>Package</i>	12
-----------	------------------------------	----



DAFTAR GAMBAR

Gambar 2.1	Ruang lingkup keterhubungan J2EE, J2SE, dan J2ME	6
Gambar 2.2	Lingkup Configuration	7
Gambar 2.3	Hubungan J2ME dan J2SE	7
Gambar 2.4	Arsitektur J2ME	8
Gambar 2.5	Alur hidup MIDlet	10
Gambar 3.1	Rancangan Alur Program	15
Gambar 3.2	Jendela KToolbar	17
Gambar 3.3	New Project	18
Gambar 3.4	Open Project	19
Gambar 3.5	Setting	22
Gambar 3.6	MIDlets pada <i>Setting</i>	22
Gambar 3.7	<i>Use Case Diagram</i>	23
Gambar 3.8	<i>Class Diagram</i>	23
Gambar 3.9	<i>Sequence Diagram</i> Mudharabah	24
Gambar 3.10	<i>Sequence Diagram</i> Murabahah	26
Gambar 3.11	<i>Sequence Diagram</i> Help	28
Gambar 3.12	Rancangan Menu Awal	29
Gambar 3.13	Rancangan <i>Interface</i> Mudharabah	30
Gambar 3.14	Rancangan <i>Interface</i> Menu Save	35
Gambar 3.15	Rancangan Menu Murabahah	35
Gambar 3.16	Rancangan <i>Interface</i> Petunjuk	35
Gambar 4.1	Cara Pengisian <i>Form</i> Mudharabah	35

Gambar 4.2	Hasil Perhitungan Mudharabah	35
Gambar 4.3	Fungsi Simpan	35
Gambar 4.4	Cara Pengisian <i>Form</i> Murabahah	35
Gambar 4.5	Hasil Perhitungan Murabahah	35
Gambar 4.6	Petunjuk Syariah	35
Gambar 4.7	Pesan Kesalahan Input Nisbah dan Margin	35
Gambar 4.8	Pesan Kesalahan Nisbah dan Margin Melebihi Standar	35
Gambar 4.9	Pesan Kesalahan Pada Input Modal	35
Gambar 4.10	Pesan Kesalahan Jangka Waktu	35
Gambar 4.11	Pesan Kesalahan Penghasilan bulan ini	35
Gambar 4.12	Pesan Kesalahan Petunjuk	35



ABSTRAK

Setiap orang yang ingin meminjam uang baik itu untuk modal usaha atau untuk kebutuhan pribadi pasti akan meminjam ke bank. Meminjam uang di bank umum sudah biasa dilakukan, tapi meminjam uang di bank syariah hanya digunakan sebagian orang yang tahu tentang bank syariah. Perbedaan dari meminjam uang dari bank umum atau syariah terletak dari riba yang ditentukan. Untuk memudahkan nasabah mengetahui seperti apa pinjaman syariah itu dapat menggunakan *handphone* (HP). Oleh karena itulah aplikasi informasi pinjaman syariah via *handphone* ini dibuat. Untuk proses perancangan akan digunakan software aplikasi J2ME dan menggunakan KToolbar. Perancangan perangkat lunak *handheld devices* dirancang berdasarkan tampilan – tampilan yang dibutuhkan. Tampilan tersebut terdiri dari menu – menu yang akan mempermudah *user* dalam pencarian informasi pinjaman syariah. Penggunaannya sangat mudah hanya dengan memilih jenis pinjaman apa yang diinginkan. Setelah itu dimasukkan input berupa angka dan kemudian dengan menekan tombol hitung maka output secara otomatis akan keluar hasil perhitungan dari input tersebut. Digunakannya teknologi J2ME (Java 2 Micro Edition) karena teknologi ini bersifat bebas platform maupun perangkat yang digunakan. Selain itu memori yang digunakan sangat kecil, sehingga tidak begitu menghabiskan memori pada *handphone* yang ada dan proses instalasi begitu mudah.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemanfaatan peralatan elektronik dapat diaplikasikan dalam berbagai macam hal. Berbagai hal tersebut antara lain pemanfaatan peralatan elektronik dalam bidang informasi. Dalam bidang informasi perangkat elektronik dapat digunakan untuk menampilkan berbagai macam informasi, salah satunya yaitu informasi pinjaman. Informasi pinjaman ini merupakan sebuah program java yang digunakan pada *handphone* berplatform java, sedangkan untuk membuat program digunakan sebuah software J2ME (Java 2 Micro Edition) keluaran dari SUN Microsystems. Program ini akan disusun sesuai dengan bahasa java yang sering digunakan pada umumnya, program informasi pinjaman akan disimulasikan pada *handphone* Sony Ericsson W810i.

Dengan adanya informasi pinjaman pada *handphone* berbasis java maka akan mempermudah dalam memahami perhitungan pinjaman tanpa harus berkonsultasi langsung ke bank. Dari hasil pengamatan inilah dapat dibuat suatu pembuatan aplikasi “Informasi Pinjaman Perbankan Syariah Berbasis Java dengan J2ME pada Handphone”.

1.2 Rumusan Masalah

Dari latar belakang dan dasar pemikiran diatas, maka dapat diambil rumusan masalah sebagai berikut: “Bagaimana cara membuat aplikasi informasi pinjaman

perbankan syariah dengan menggunakan bahasa pemrograman java pada *handphone* dengan menggunakan software J2ME (Java 2 Micro Edition)".

1.3 Batasan Masalah

Dengan adanya batasan masalah maka dapat untuk menyederhanakan dan mengarahkan penelitian dan pembuatan sistem agar tidak menyimpang dari apa yang akan diteliti dan dikembangkan. Batasan – batasan masalahnya adalah sebagai berikut :

1. Penulisan program menggunakan bahasa java.
2. Menggunakan software J2ME (Java 2 Micro Edition).
3. Menggunakan J2SE SDK versi 1.4.2 atau lebih tinggi.
4. Menggunakan J2ME Wireless Toolkit versi Sony Ericsson.
5. Mensimulasikan pada *platform handphone* Sony Ericsson W810i.
6. K-Toolbar sebagai pembuat *project*.

1.4 Tujuan Perancangan

Berkembangnya sistem informasi pada *handphone* memberikan kemudahan dari para pengembang aplikasi untuk menciptakan terobosan – terobosan yang berguna untuk para pengguna. Tujuan perancangan aplikasi informasi pinjaman syariah adalah memberikan kemudahan kepada pengguna untuk mengetahui perhitungan – perhitungan pinjaman dengan cara mensimulasikan pada *handphone*.

I.5 Manfaat perancangan

Aplikasi tersebut memberikan manfaat yang sangat besar dalam hal pemahaman perhitungan pinjaman pada bank terutama bank syariah. Sehingga pengguna dapat mengetahui bagaimana dan seperti apa pinjaman uang untuk kepentingan konsumtif (pribadi) atau untuk modal usaha melalui bank syariah. Pengguna cukup dengan mensimulasikan melalui *handphone*, tanpa harus menghitung dengan susah payah. Aplikasi ini merupakan simulasi yang hampir mencakup pinjaman yang tersedia pada setiap bank – bank syariah yang ada saat ini.

I.6 Sistematik Penulisan Laporan

Untuk memudahkan sistematika penulisan laporan, maka dibagi dalam beberapa bab pembahasan dengan urutan sebagai berikut :

BAB I PENDAHULUAN

Bab ini akan menguraikan latar belakang dan alasan pemilihan judul, tujuan penulisan, pembatasan masalah, metodologi, dan sistem penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang java, J2ME, & *software* yang digunakan sebagai dasar proses perancangan aplikasi.

BAB III PERANCANGAN SOFTWARE

Bab ini berisi penjelasan tentang diagram blok dan proses pembuatan aplikasi yang dibuat.

BAB IV PENGUJIAN, ANALISA DAN PEMBAHASAN

Bab ini menguraikan proses pengujian, analisa dan pengamatan dari aplikasi yang dibuat.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran – saran sebagai masukan untuk penunjang pembuatan aplikasi tersebut.



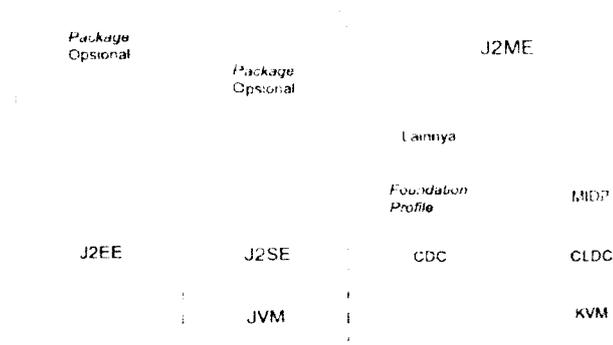
BAB II

LANDASAN TEORI

2.1 Definisi Java

Java menurut definisi dari SUN adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari Java *platform* (generasi awalnya adalah Java Development Kit). Java berdiri di atas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file .class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa java merupakan bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi terdapat JVM.

Platform Java memiliki tiga buah edisi yang berbeda, yaitu J2EE (Java 2 Enterprise Edition), J2ME (Java 2 Micro Edition), dan J2SE (Java 2 Second Edition). J2EE adalah kelompok dari beberapa API dari java dan teknologi selain java. J2EE sering dianggap sebagai *middleware* atau teknologi yang berjalan di *server*. Selain itu J2EE juga mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah sistem informasi. Sedangkan J2SE adalah lingkungan dasar dari java. Ruang lingkup keterhubungan J2EE, J2SE, dan J2ME dapat dilihat pada Gambar 2.1.

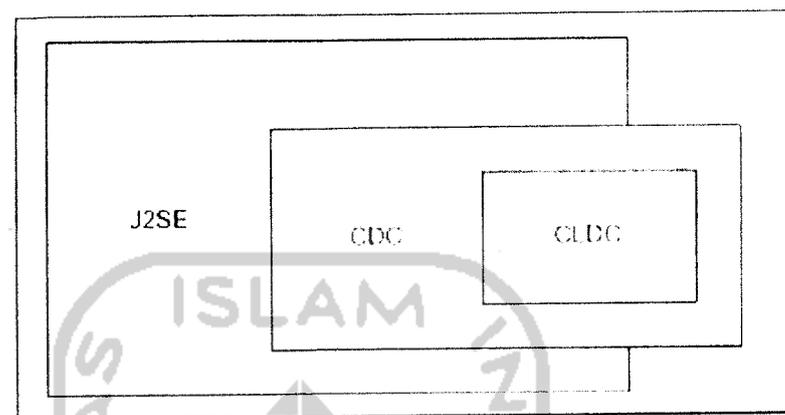


Gambar 2.1 Ruang lingkup keterhubungan J2EE, J2SE, dan J2ME

2.2 Pengertian J2ME

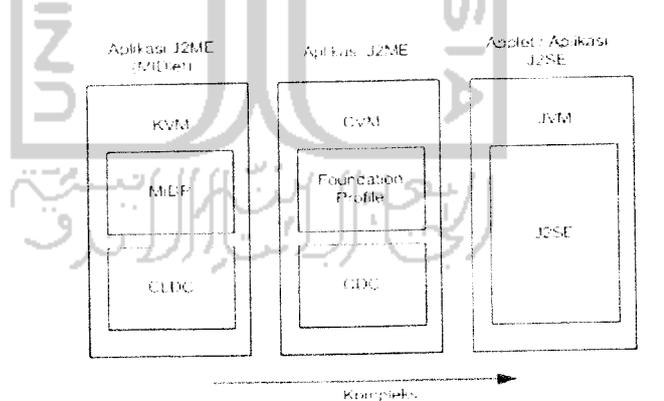
Java 2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. J2ME membawa java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop* yang biasanya lebih kecil dibandingkan perangkat komputer *desktop*. J2ME biasa digunakan pada telepon selular, pager, PDA, dan sejenisnya. Teknologi J2ME juga memiliki keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat J2ME (*device*) yang digunakan, bisa dari segi merek ponsel, maupun kemampuan ponsel, dan dukungan terhadap J2ME. Selain itu keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebagian besar ponsel tidak mengizinkan aplikasi J2ME menulis pada *file* karena alasan keamanan. Dalam J2ME terdapat dua buah jenis *konfigurasi* dan *profile*, yaitu untuk *konfigurasi* terdapat CLDC (*Connected Limited Device Configuration*) untuk perangkat kecil dan CDC (*Connected Device Configuration*)

untuk perangkat yang lebih besar. Lingkup CLDC dan CDC dapat dilihat pada Gambar 2.2.



Gambar 2.2 Lingkup Configuration

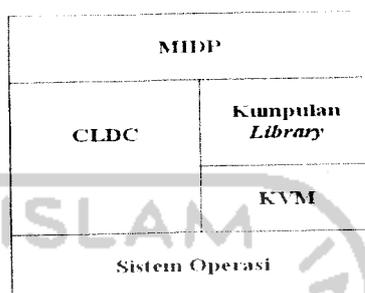
Sedangkan pada profile, yaitu MIDP dan *Foundation Profile*. Disini dapat dilihat pada Gambar 2.3 keterhubungan antara *configurasi* dan *profile* yang ada pada J2ME beserta jenis mesin Virtualnya.



Gambar 2.3 Hubungan J2ME dan J2SE

2.2.1 Bagian – Bagian Arsitektur J2ME

Pada bagian ini dapat dijelaskan bagian – bagian yang terdapat pada J2ME dan letak arsitektur J2ME ditunjukkan pada Gambar 2.4.



Gambar 2.4 Arsitektur J2ME

- a. CLDC (*Connected Limited Device Configuration*) adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa library dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada *handphone*, pager, dan PDA. Spesifikasi pada J2ME adalah spesifikasi minimal dari *package*, kelas, dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat – alat tersebut. JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*).
- b. MIDP (*Mobile Information Device Profile*) adalah spesifikasi untuk sebuah profile J2ME. API tambahan untuk daur hidup (*life cycle*) aplikasi antarmuka, jaringan, dan penyimpanan persisten. Daur hidup (*life cycle*) dari sebuah MIDlet ditangani oleh AMS (*Application Management Software*). AMS ini adalah sebuah

lingkungan tempat siklus dari sebuah MIDlet mampu diciptakan, dijalankan, dihentikan, maupun dihilangkan. AMS juga sering dinamakan JAM (Java Application Manager). Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan MIDP 2.0 dibanding 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). MIDP *User Interface* API memiliki API level tinggi dan level rendah. API level rendah berbasiskan penggunaan dari kelas abstrak *Canvas*, sedangkan kelas API level tinggi antara lain *Alert*, *Form*, *List*, dan *TextBox* yang merupakan ekstensi dari kelas abstrak *Screen*.

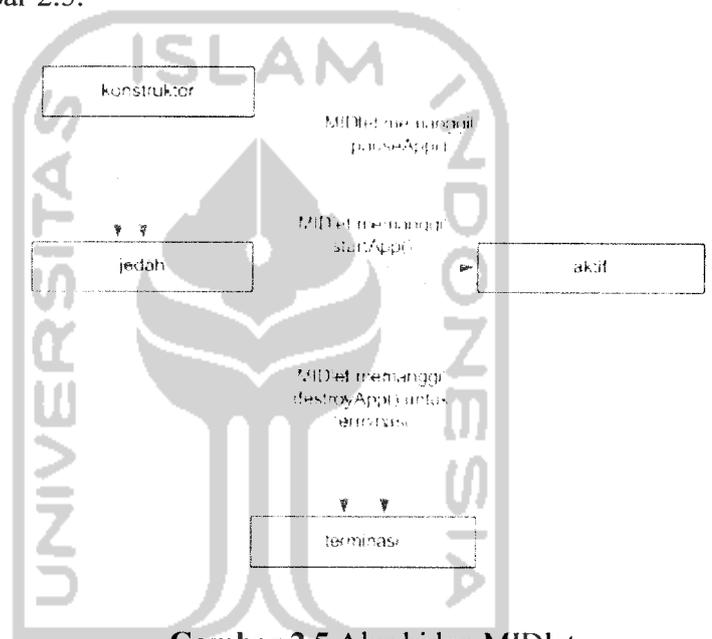
- c. KVM (*Kilobyte Virtual Machine*) adalah paket JVM yang didesain untuk perangkat yang kecil. KVM mendukung sebagian dari fitur – fitur JVM, seperti misalnya KVM tidak mendukung operasi *floating-point* dan finalisasi objek.

2.2.2 MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas *javax.microedition.midlet.MIDlet* yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan sub kelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.

a. Daur Hidup MIDlet

MIDlet terdiri dari beberapa metode yang harus ada yaitu *constructor ()*, *protected void startApp () throws MIDletStateChangeException*, *protected void destroyApp (boolean unconditional) throws MIDletStateChangeException*. Alur hidup MIDlet dapat dilihat pada Gambar 2.5.



Gambar 2.5 Alur hidup MIDlet

Ketika MIDlet dijalankan maka akan diinisialisasi dengan kondisi pause dan dijalankan *pauseApp ()*, kondisi berikutnya adalah fungsi MIDlet dijalankan, yaitu pada *startApp ()*. Metode yang ada tersebut diimplementasikan sebagai *protected*, hal ini dimaksudkan agar MIDlet lain tidak dapat memanggil metode tersebut. Pada saat pemakai keluar dari MIDlet, maka metode *destroyApp ()* akan dijalankan sebelum MIDlet benar – benar tidak berjalan lagi. Metode *notifyDestroyed ()* akan

dipanggil sebelum MIDlet benar – benar tidak berjalan lagi *destroyApp ()* akan memanggil *notifyDestroyed ()*, dan *notifyDestroyed ()* akan memberitahu *platform* untuk MIDlet menterminasi dan membersihkan semua sumber daya pada MIDlet.

Dalam implementasinya, MIDlet memiliki struktur direktori sebagai berikut :

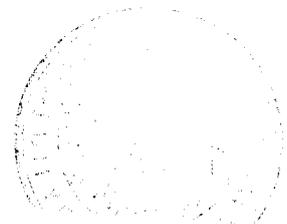
- **src**
menyimpan *source code* untuk MIDlet dan kelas lain yang diperlukan.
- **res**
menyimpan sumber daya yang dibutuhkan oleh MIDlet, seperti misalnya gambar icon.
- **lib**
menyimpan file JAR atau ZIP yang berisi library tambahan yang dibutuhkan MIDlet.
- **bin**
menyimpan file JAR, JAD, dan *manifest* yang berisi muatan komponen MIDlet.

b. Atribut MIDlet

MIDlet juga memiliki atribut *package* yang dapat dilihat pada tabel 2.1 :

Tabel 2.1 Atribut *Package*

Nama Atribut	Nilai dan Fungsi
MIDlet-Name	Nama MIDlet yang juga sebagai nama untuk file JAR-nya. Misalnya WirelessTrader
MIDlet-Version	Nomor versi dari MIDlet yang juga sebagai nomor versi pada file JAR-nya. Misalnya 1.0.1
MIDlet-Vendor	Nama provider MIDlet. Misalnya Wireless Java Inc.
MIDlet- <i>n</i>	Atribut yang mendeskripsikan MIDlet. Nilai <i>n</i> diganti dengan nilai numerik dimulai dari angka 1. Format nilai dari atribut ini terkait dengan atribut yang dideskripsikan selanjutnya.
MicroEdition-Profile	Versi dari spesifikasi MIDlet yang dapat berjalan. Dapat lebih dari satu versi yang dipisah dengan spasi
MicroEdition-Configuration	Konfigurasi J2ME yang dibutuhkan untuk MIDlet.
MIDlet-Description	Deskripsi MIDlet. Misalnya MIDlet untuk koneksi Internet.
MIDlet-Icon	Icon yang digunakan MIDlet. Misalnya wireless.png
MIDlet-Info-URL	URL dan file yang berisi informasi mengenai MIDlet. Misalnya http://www.wireless.com/info.html
MIDlet-Data-Size	Nilai minimum dari besarnya tempat penyimpanan persisten yang dibutuhkan (dalam <i>byte</i>). Misalnya 512.
MIDlet-Jar-URL	URL dan file JAR. Misalnya http://www.wireless.com/MIDlet.jar
MIDlet-Jar-Size	Ukuran file JAR dalam <i>byte</i> . Misalnya 10312.
MIDlet-Install-Notify	Sebuah URL yang digunakan untuk melaporkan sukses atau gagalnya penginstalan MIDlet dari <i>Remote Server</i> .
MIDlet-Delete-Confirm	Pesan yang ditampilkan ke pemakai sebelum MIDlet dihapus dari alat dimana MIDlet diinstal.
MIDlet-specificattributes	Pengembangan MIDlet dapat menyediakan konfigurasi minimum untuk MIDlet dengan memasukkan atribut ini.



2.2.3 JAD (*Java Application Descriptor*)

Digunakan untuk mendeskripsikan isi aplikasi untuk keperluan pemetaan. File JAD berisi deskripsi file JAR (*Java Archive*) dan pemetaan atribut MIDlet, sedangkan file JAR berisi kumpulan kelas dan *resource*.

2.3 Pengertian Bank

Bank berasal dari kata *banque* dalam bahasa Prancis, dan dari *banco* dalam bahasa Italia, yang dapat berarti peti/lemari atau bangku. Kata peti atau lemari menyiratkan fungsi sebagai tempat menyimpan benda – benda berharga, seperti peti emas, peti berlian, peti uang, dan sebagainya. Fungsi dasar bank adalah :

1. Tempat untuk menitipkan uang dengan aman (*safe keeping function*).
2. Menyediakan alat pembayaran untuk membeli barang dan jasa (*transaction function*).

Bank memiliki peranan yang sangat penting dalam perekonomian dewasa ini. Sebagai lembaga intermediasi, bank konvensional menerima simpanan dari nasabah dan meminjamkannya kepada nasabah (unit ekonomi) lain yang membutuhkan dana. Atas simpanan para nasabah itu bank memberi imbalan berupa bunga. Demikian pula, atas pembelian pinjaman itu bank mengenakan bunga kepada para peminjam.

2.4 Pengertian Bank Syariah

Bank syariah didirikan dengan tujuan untuk mempromosikan dan mengembangkan penarapan prinsip – prinsip Islam, syariah dan tradisinya ke dalam transaksi keuangan dan perbankan serta bisnis lain yang terkait. Prinsip utama yang diikuti oleh bank syariah itu adalah :

- a. Larangan riba dalam berbagai bentuk transaksi.
- b. Melakukan kegiatan usaha dan perdagangan berdasarkan perolehan keuntungan yang sah.
- c. Memberikan zakat.

Sepanjang praktek perbankan konvensional tidak bertentangan dengan prinsip – prinsip Islam, bank – bank syariah telah mengadopsi sistem dan prosedur perbankan yang ada. Apabila terjadi pertentangan dengan prinsip – prinsip syariah Islam, dewan syariah yang akan mengurusnya. Dewan syariah berfungsi memberikan saran kepada perbankan Islam guna memastikan bahwa bank Islam tidak terlibat dalam unsur – unsur yang tidak disetujui oleh Islam.

2.4.1 Produk Pembiayaan Bank Syariah

Pada bank syariah terdapat jenis pembiayaan yang diberikan kepada nasabah yang membutuhkan dana untuk membeli kebutuhan pribadi (konsumtif) atau untuk modal usaha. Jenis – jenis pembiayaan bank syariah antara lain :

a. Pembiayaan Mudharabah

Pembiayaan Mudharabah adalah pembiayaan dimana seluruh modal kerja yang dibutuhkan nasabah ditanggung oleh bank. Keuntungan yang diperoleh dibagi sesuai dengan nisbah yang telah disepakati.

b. Pembiayaan Murabahah

Pembiayaan Murabahah adalah pembiayaan berdasarkan akad jual beli antara bank dan nasabah. Bank membeli barang yang dibutuhkan dan menjualnya kepada nasabah sebesar harga pokok ditambah dengan keuntungan margin yang disepakati. Selain membantu nasabah dalam membeli barang – barang usaha, dapat juga digunakan untuk membeli berbagai macam hal seperti mobil, rumah, dan pendidikan.

Dalam jenis pembiayaan prinsip yang digunakan hampir sama yaitu dengan menggunakan akad yang telah disepakati bersama – sama.

2.4.2 Perhitungan Peminjaman

Keuntungan yang didapat dari pembiayaan berupa nisbah (bagi hasil) untuk pembiayaan mudharabah. Sedangkan untuk pembiayaan murabahah berupa margin yang telah ditetapkan pada akad yang telah disepakati. Klasifikasi perhitungan dapat dilihat dibawah ini.

1) Pembiayaan Mudharabah

- a. Dana yang dibutuhkan.
- b. Jangka waktu.
- c. Nisbah bagi hasil (akad) pada laba bersih.
- d. Biaya administrasi
- e. Pembayaran bagi hasil setiap bulan.
- f. Pengembalian pokok dapat dilakukan pada akhir periode atau setiap bulan.

Rumus : $(\text{Penghasilan Bulan Ini} * \text{Nisbah} / 100) + (\text{Pinjaman Modal} / \text{Jangka Waktu})$

Contoh : Seorang nasabah ingin meminjan uang untuk menambah modal usaha perdagangan sebesar 150.000.000,-. Setelah dilakukan analisa keuangan, maka disetujui fasilitas Mudharabah oleh bank syariah kepada nasabah tersebut, dengan persyaratan sebagai berikut :

1. Pinjaman modal Rp. 150.000.000,-
2. Nisbah Bagi Hasil (berdasarkan laba bersih) 20% untuk bank dan 80% untuk nasabah.
3. Obyek bagi hasil yaitu laba bersih.
4. Biaya Administrasi Rp. 1.500.000,-
5. Jangka waktu 36 bulan.

6. Misal laba bersih bulan ini Rp. 5.000.000,- maka angsuran bulan ini
 : $(\text{Rp. } 5.000.000,- * 20 / 100) + (\text{Rp. } 150.000.000,- / 36)$
 : Rp. 5.166.666,-

2) Pembiayaan Murabahah

- a. Dana yang dibutuhkan.
- b. Margin ditetapkan oleh bank.
- c. $\text{Harga Jual} = \text{Harga Beli} + \text{Margin}$.
- d. Biaya administrasi.
- e. Jangka waktu pelunasan.
- f. Angsuran tiap bulan.

Rumus : $((\text{Margin} / 100 * \text{Pinjaman Uang}) + \text{Pinjaman Uang}) / \text{Jangka Waktu}$.

Contoh : Seorang nasabah ingin meminjan uang untuk membeli sebuah mobil baru yang harganya berkisar Rp. 150.000.000,-. Setelah *Account Manager* Bank Syariah mereview neraca dan laporan keuangan sumber pengembalian nasabah tersebut, maka telah disetujui permohonan fasilitas Murabahah sebagai berikut :

1. Harga beli barang dari Supplier Rp. 150.000.000,-
2. Margin Bank Syariah (20% pa. efektif) sebesar Rp. 30.000.000,-
3. Biaya Administrasi Rp. 1.500.000,-

4. Jangka waktu pelunasan 36 bulan.

5. Angsuran/Bulan : $\frac{30.000.000 + 150.000.000}{36}$

36

: Rp. 5.000.000,-/bulan.

Inilah perhitungan dasar setiap bank syariah dalam mengambil keuntungan. Untuk setiap bank margin atau nisbah memiliki nilai yang berbeda – beda, biasanya dalam bentuk persentase.



BAB III

PERANCANGAN SOFTWARE

3.1 Metode Perancangan

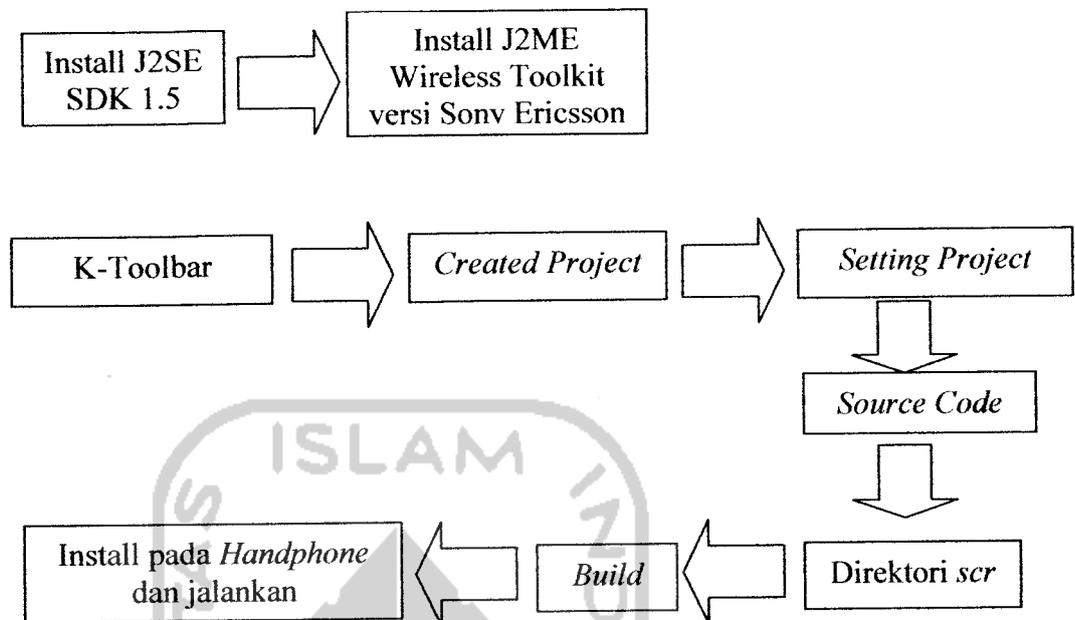
Perancangan perangkat lunak *handheld devices* dirancang berdasarkan tampilan – tampilan yang dibutuhkan. Tampilan tersebut terdiri dari menu – menu yang akan mempermudah *user* dalam pencarian informasi pinjaman syariah.

Metode perancangan yang digunakan untuk membangun Aplikasi Informasi Pinjaman Perbankan Syariah Berbasis Java dengan J2ME ini adalah menggunakan metode perancangan berarah objek dengan menggunakan bahasa UML (*Unified Modelling Language*). Bahasa ini digunakan karena merupakan bahasa yang *process-independent*, yaitu bahasa yang saling bebas terhadap proses yang ada di dalamnya. Alasan utama kenapa menggunakan tipe pemodelan UML ini karena UML dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun.

Tahapan perancangan yang akan dibahas merupakan perancangan yang akan menghasilkan kebutuhan sistem dan pemilihan teknologi, hal ini mencakup analisis dan perancangan desain.

3.2 Diagram Perancangan

Pada pembuatan aplikasi dibutuhkan beberapa *software* pendukung, dengan rancangan program sebagai berikut :



Gambar 3.1 Rancangan Alur Program

Dibawah ini akan dijelaskan masing – masing alur yang terdapat pada diagram diatas

a. Install J2SE SDK 1.5

J2SE SDK 1.5 seperti *virtual sistem* yang berfungsi untuk dapat menjalankan *platform* java pada lingkungan sistem komputer. Untuk mendapatkan *software* tersebut dapat di *download* langsung pada situs internet.

b. Install J2ME Wireless Toolkit versi Sony Ericsson

J2ME Wireless Toolkit adalah kakas yang menyediakan lingkungan *emulator*, dokumentasi beserta contoh – contoh aplikasi java untuk perangkat kecil (*small device*). Basis yang digunakan J2ME Wireless Toolkit berupa CLDC dan MIDP. J2ME Wireless Toolkit adalah program

yang meniru kerja ponsel yang mendukung MIDP atau yang disebut *emulator*. Oleh karena itu, belum tentu MIDlet yang berjalan di *emulator* juga berjalan pada ponsel yang sebenarnya, karena juga bergantung pada kemampuan dan kapasitas ponsel yang digunakan. Untuk *emulator* yang digunakan merupakan versi dari *handphone* Sony Ericsson W810i.

c. K-Toolbar

K-Toolbar memiliki peranan penting pada J2ME, peranan tersebut adalah berfungsi sebagai pembuat *project* dan segala jenis *setting* yang akan digunakan.

d. Created Project

Berfungsi untuk membuat *project* yang akan dibuat, setelah itu akan muncul beberapa direktori – direktori baru yang berhubungan dengan program ini. Direktori – direktori tersebut adalah bin, lib, res, src, project.property.

e. Setting Project

Setting project aplikasi J2ME dapat diset pada K-Toolbar. Jendela *setting* biasanya dapat dibuka pada saat membuat *project* maupun ketika sedang mengerjakan *project*. Jika menggunakan *setting* standar cukup tekan OK. Jika ingin menyesuaikan *setting* dengan kebutuhan aplikasi maka *setting* dapat diset sendiri.

f. Source Code

Merupakan tulisan – tulisan program yang ditulis sesuai dengan aplikasi yang akan dibuat. *Source code* biasanya berupa *text* biasa yang kemudian dimasukkan pada direktori tertentu.

g. Direktori scr

Lokasi penempatan file – file *source code* pada aplikasi yang berupa *text*.

h. Build

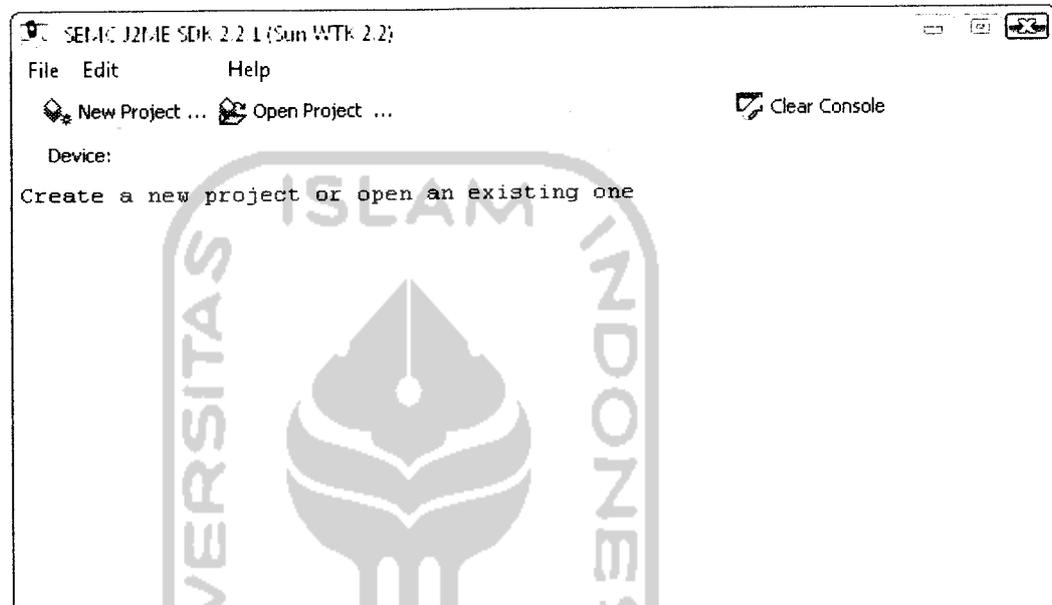
Build digunakan untuk menjalankan *emulator*, dengan cara mengkompilasikan *source code*. *Build* akan mengalami keberhasilan ketika pada layar K-Toolbar muncul tulisan *build complete*, jika *source code* gagal dikompilasi maka yang keluar pada layar K-Toolbar adalah pesan – pesan kesalahan. Jika hal ini terjadi, perbaiki kesalahan pada *source code* dan lakukan kompilasi kembali hingga *source code* lulus kompilasi.

i. Install pada Handphone dan Jalankan

Setelah semua berjalan lancar dan sudah bisa berjalan pada *emulator*. Tinggal melakukan proses terakhir dengan menjalankan secara langsung pada *handphone*.

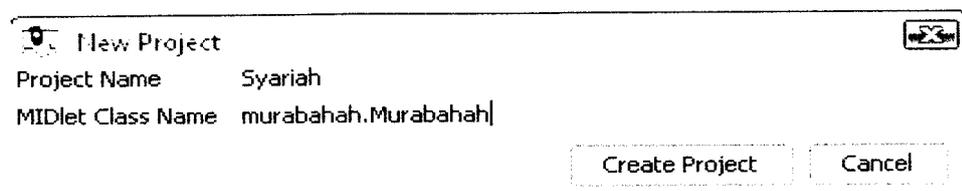
3.3 Membuat dan Membuka Aplikasi

Untuk membuka sebuah aplikasi J2ME, terlebih dahulu harus meng-install JDK (*Java Development Kit*) dan Java Wireless Toolkit pada komputer. Kemudian buka KToolbar hingga muncul seperti pada Gambar 3.2.



Gambar 3.2 Jendela KToolbar

Untuk membuat proyek aplikasi J2ME klik tombol New Project hingga muncul jendela seperti pada Gambar 3.3. Isikan nama proyek dan nama kelas MIDlet lalu klik Create Project, nama kelas diisi harus merupakan nama turunan dari MIDlet (*extend MIDlet*). Disini akan dibuat *project* Syariah dan akan diikuti kelas turunan Murabahah.

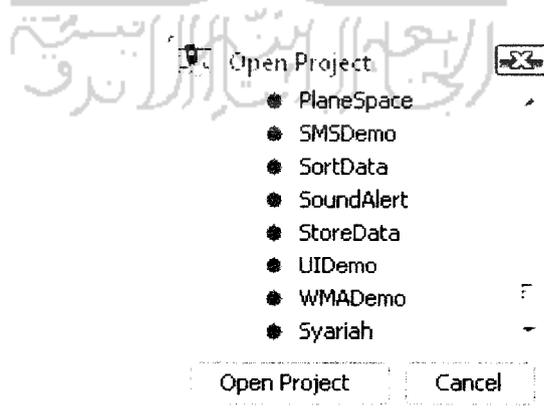


Gambar 3.3 New Project

Setelah proses membuat proyek selesai maka pada direktori *install/apps* atau *C : /WTK22/apps* akan terbentuk sebuah direktori dengan nama sesuai dengan proyek yang baru saja dibuat. Dalam direktori tersebut akan terdapat direktori – direktori sebagai berikut :

- a. Direktori *bin* yang merupakan tempat file MANIFEST.MF, file JAD, file JAR.
- b. Direktori *lib* untuk meletakkan file – file *library* (biasanya berupa file JAR).
- c. Direktori *res* untuk meletakkan file – file *resource* seperti file gambar maupun suara.
- d. Direktori *src* untuk meletakkan file – file *source code*.
- e. File *project.properties* yang menyimpan setting MIDlet.

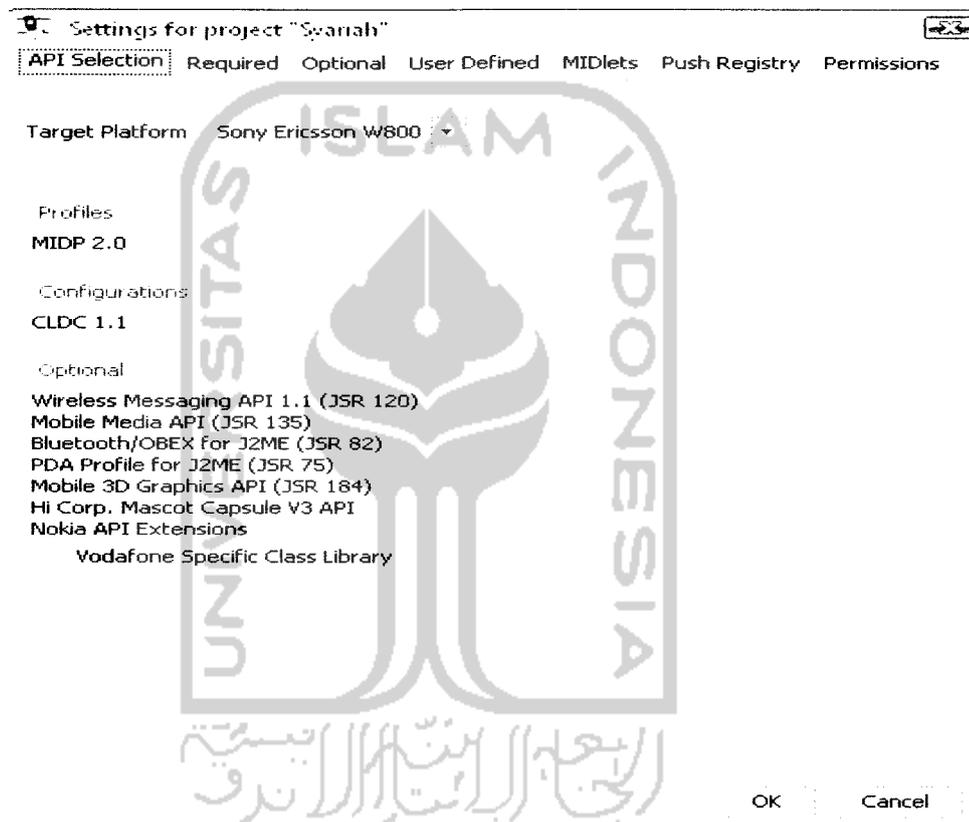
Untuk membuka sebuah proyek, klik Open Project pada KToolbar hingga muncul jendela seperti pada Gambar 3.4. pilih proyek yang akan dibuka, kemudian klik Open Project.



Gambar 3.4 *Open Project*

3.4 Setting Project

Setting proyek aplikasi J2ME dapat diset pada KToolbar. Jendela *Setting* biasanya dapat dibuka pada saat membuat proyek maupun ketika sedang mengerjakan proyek. Klik tombol *Setting* hingga muncul jendela seperti pada Gambar 3.5.

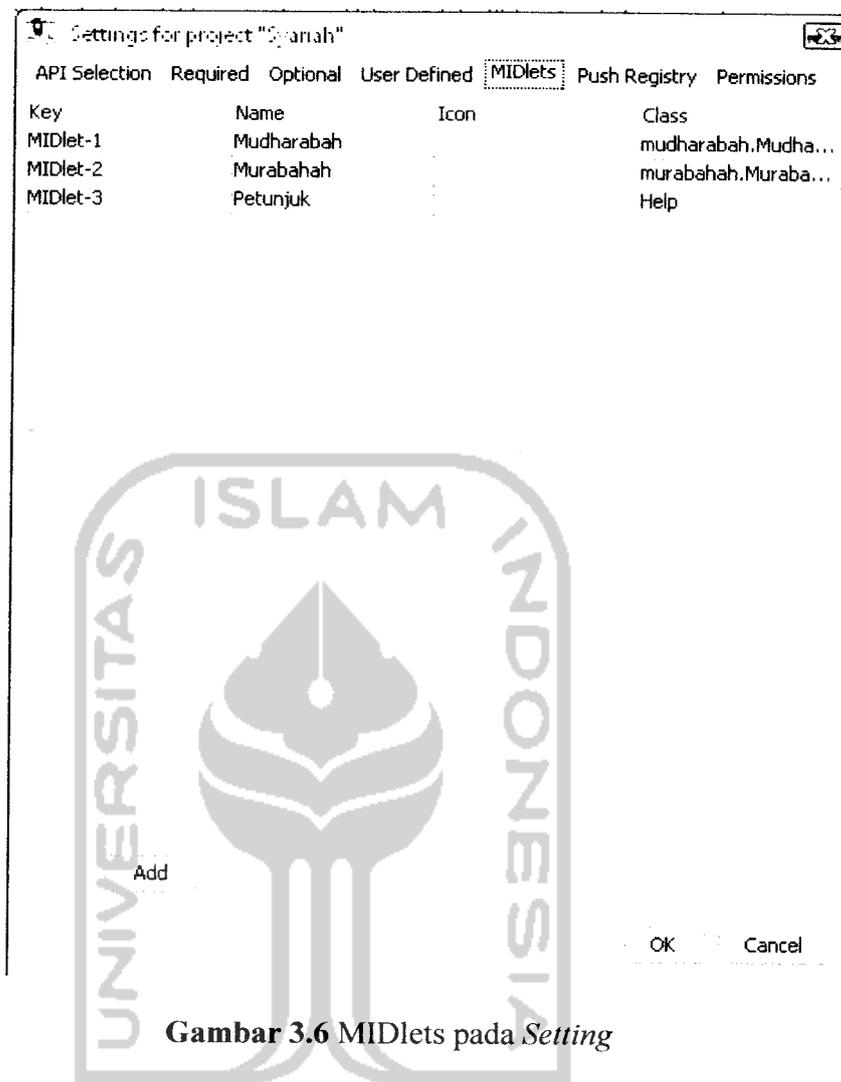


Gambar 3.5 *Setting*

Jika ingin menggunakan *Setting* standar maka jika jendela *Setting* muncul langsung saja tekan tombol OK. Jika ingin menyesuaikan *Setting* dengan kebutuhan aplikasi, maka *Setting* juga dapat diset sendiri. Pada tab API Selection pelayanan yang dibutuhkan untuk MIDlet yang dibuat disesuaikan dengan SE W800.

Dalam perancangan aplikasi syariah *Setting* yang digunakan disesuaikan dengan *Setting* yang telah ditentukan oleh W800. Perbedaan dari *Setting* standar dengan W800 terletak pada pemilihan *optional* dan *configurations* menggunakan CLDC 1.1. Sebenarnya banyak *handphone* yang telah mendukung CLDC 1.0, sebaliknya hanya *handphone* tertentu yang mendukung CLDC 1.1. Perbedaan CLDC 1.0 dan CLDC 1.1 adalah pada penanganan *floating point*, sedangkan pada CLDC 1.0 belum ada. Penanganan *floating point* banyak digunakan pada aplikasi yang menggunakan angka – angka untuk perhitungan.

Tab MIDlets seperti Gambar 3.6, digunakan untuk menambah *class* pada *package* Syariah. Pada penambahan *class* langsung terdapat direktori baru pada scr yang disesuaikan dengan nama – nama *class* yang ditambahkan. Untuk menambah *class* tekan tombol *Add*, kemudian masukan nama *class* yang diinginkan dan masukkan juga turunan *class* dari MIDlet tersebut setelah semua kotak yang kosong kemudian tekan tombol OK. Kemudian secara otomatis penambahan *class* tersebut akan disimpan langsung pada direktori scr.



Gambar 3.6 MIDlets pada *Setting*

3.5 Hasil Perancangan

Perancangan aplikasi informasi pinjaman syariah yang diakses melalui *handphone* berbasis java dengan teknologi J2ME, hanya terdiri dari satu tahap perancangan. Tahap perancangan itu adalah :

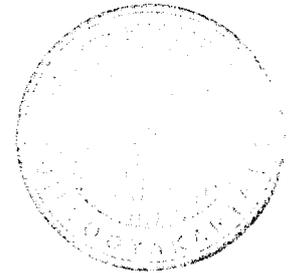
1. Perancangan Aplikasi Informasi Pinjaman Syariah via *Handphone*.

3.6 Perancangan Aplikasi

Penggunaan UML yang merupakan bahasa yang *process-independent* yaitu bahasa yang saling bebas terhadap proses yang ada di dalamnya. Dalam

perancangan aplikasi informasi pinjaman syariah via *handphone* ini terbagi dalam berbagai sudut pandang (*view*) yang dapat di kategorikan ke dalam diagram berikut :

1. *Use Case Diagram*.
2. *Class Diagram*.
3. *Sequence Diagram*.



3.6.1 *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Satu hal yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dan sistem. *Use case* merupakan pekerjaan tertentu seorang aktor dan sebuah aktor adalah sebuah manusia atau sistem yang terotomasi lain yang berinteraksi dengan sistem yang melakukan pekerjaan tersebut.

Elemen diagram *use case* adalah :

1. Aktor
Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu “siapa” atau “apa” yang menggunakan sistem. Sedangkan yang dimaksud dengan berinteraksi adalah aktor mengirim atau menerima pesan ke atau dari sistem, atau mempertukarkan informasi dengan sistem. Aktor yang terlibat adalah

pemakai aplikasi informasi pinjaman syariah itu sendiri (selanjutnya disebut *user*) yang menganalisis ke sistem.

2. *Use Case* Mudharabah

User dapat menghitung pinjaman yang bersifat untuk kebutuhan usaha, baik perorangan atau perusahaan. Sistem kemudian memunculkan masukan *input* pinjaman modal, jangka waktu, nisbah, dan penghasilan bulan ini. Kemudian *user* tinggal menekan tombol hitung sehingga langsung keluar *output* yang berupa angsuran bulan ini dan biaya administrasi.

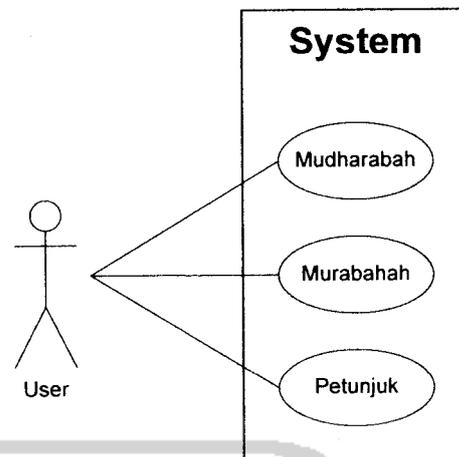
3. *Use Case* Murabahah

User dapat menghitung pinjaman yang bersifat untuk kebutuhan konsumtif (pribadi), baik perorangan atau perusahaan. Sistem kemudian memunculkan masukan *input* pinjaman modal, jangka waktu, dan margin. Kemudian *user* tinggal menekan tombol hitung sehingga langsung keluar *output* yang berupa angsuran bulan ini dan biaya administrasi.

4. *Use Case* Petunjuk

User dapat mengetahui informasi keterangan dan fungsi dari pinjaman mudharabah dan pinjaman murabahah.

Hubungan antara kasus dan aktor di dalamnya ditunjukkan oleh Gambar 3.7.



Gambar 3.7 Use Case Diagram

3.6.2 Class Diagram

Class Diagram yang dibangun dalam perangkat lunak akan dikelompokkan dalam satu paket (*package*) yaitu *Package SYARIAH*. *Class – class* tersebut antara lain :

1. *Class* MIDlet

Class - class yang berisi dari *class* Mudharabah, Murabahah, Help.

2. *Class* Mudharabah

Class untuk menghitung pinjaman Mudharabah (Usaha)

3. *Class* Murabahah

Class untuk menghitung pinjaman Murabahah (konsumtif)

4. *Class* Preferences

Class untuk settingan *save* pada *class* Mudharabah.

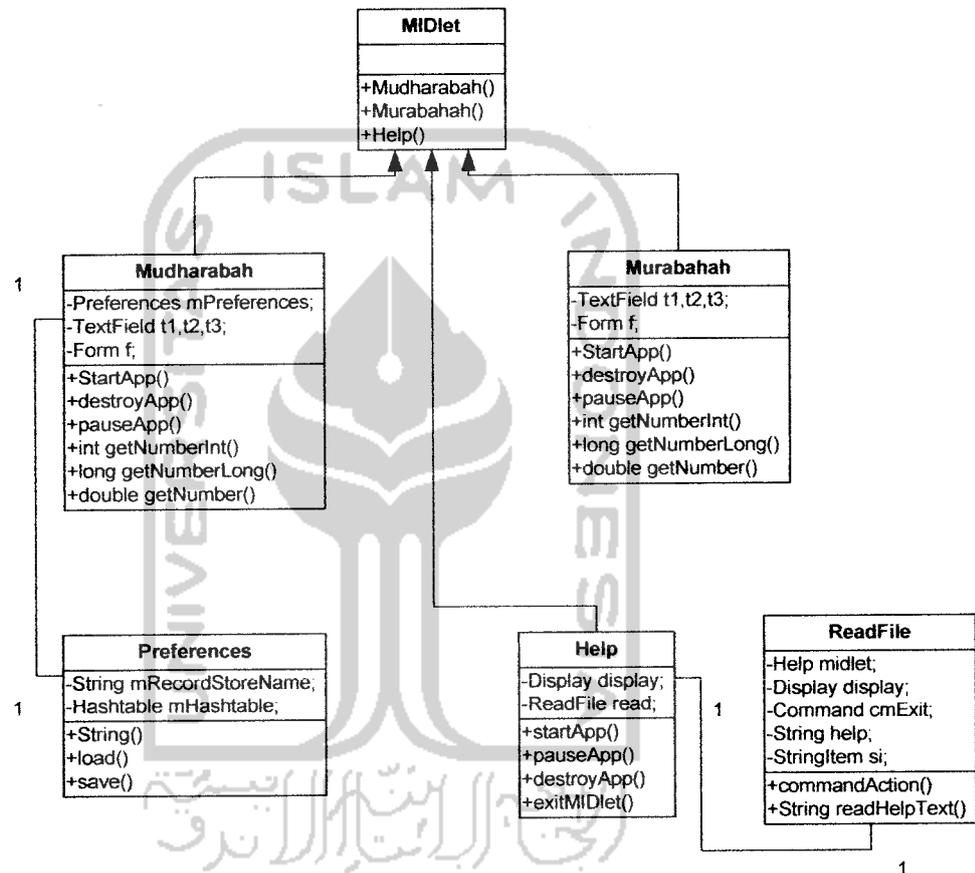
5. *Class* Help

Class untuk menampilkan Informasi Petunjuk berupa *text*, yang sudah diolah oleh *class* ReadFile.

6. Class ReadFile

Untuk membaca file *text* pada *res* dan merubahnya menjadi *string* sehingga dapat ditampilkan pada *class* Help.

Hubungan antara *class* dan *package* di atas ditunjukkan oleh Gambar 3.8 :

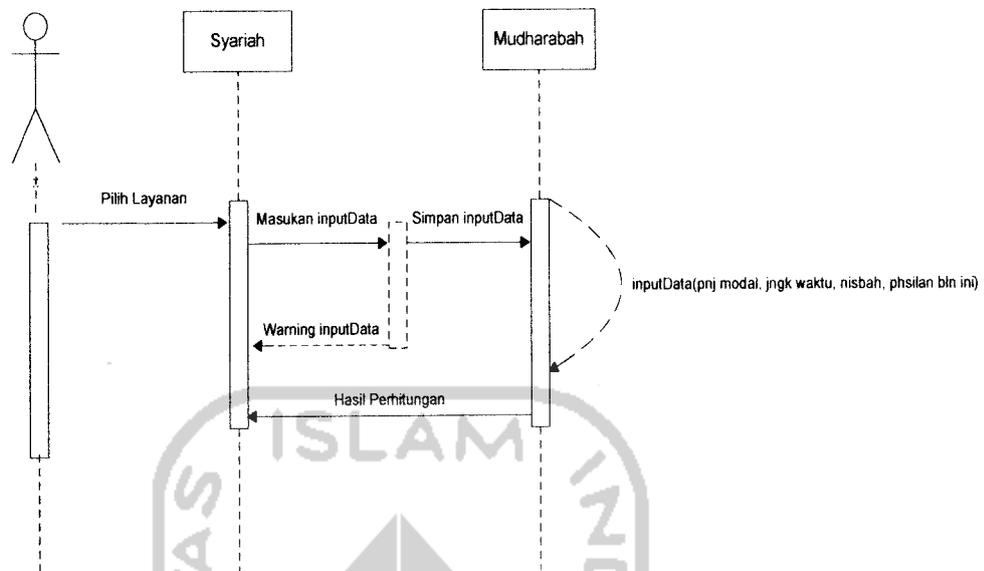


Gambar 3.8 Class Diagram

3.6.3 Sequence Diagram

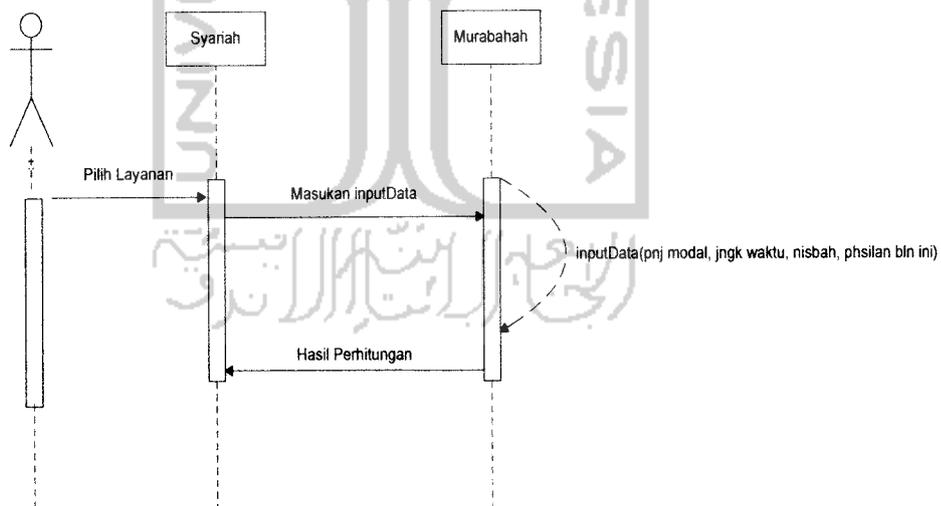
Sequence Diagram menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari objek – objek yang ada pada aplikasi Informasi Pinjaman Syariah, yaitu *Sequence Diagram* Mudharabah, *Sequence Diagram* Murabahah, dan *Sequence Diagram* Help.

Sequence Diagram Mudharabah ditunjukkan oleh Gambar 3.9 :



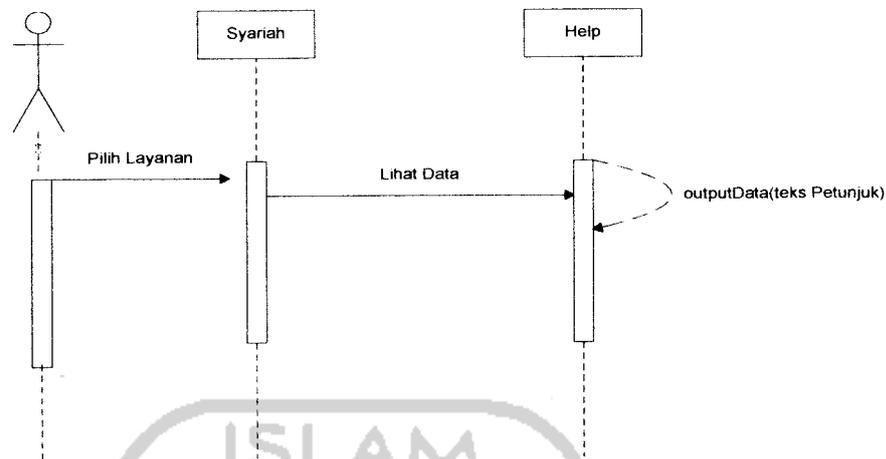
Gambar 3.9 *Sequence Diagram* Mudharabah

Sequence Diagram Murabahah ditunjukkan oleh Gambar 3.10 :



Gambar 3.10 *Sequence Diagram* Murabahah

Sequence Diagram Help ditunjukkan oleh Gambar 3.11 :



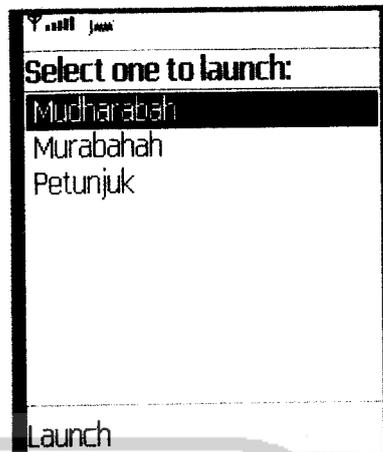
Gambar 3.11 *Sequence Diagram Help*

3.7 Perancangan Antarmuka

Dalam perancangan antarmuka Informasi Pinjaman Syariah hanya terdapat satu antarmuka, yaitu antarmuka untuk *user*. Perancangan antarmuka untuk *user* akan diakses oleh *user* dengan tingkat kemampuan yang berbeda, untuk itu diperlukan antarmuka yang mudah dimengerti dan mudah digunakan oleh pengguna atau *user*, berikut adalah perancangan antarmuka untuk *user*.

3.7.1 Rancangan Menu Awal

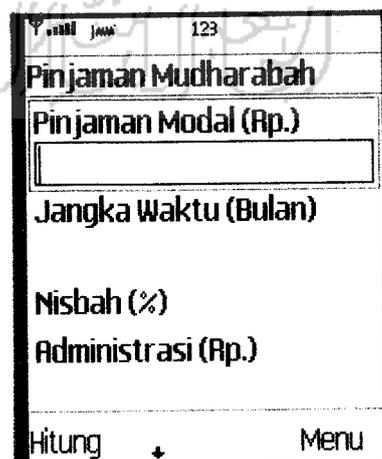
Menu awal adalah menu yang menampilkan tampilan awal dari aplikasi yang diakses oleh *user*. Rancangan menu awal ditampilkan pada Gambar 3.12.



Gambar 3.12 Rancangan Menu Awal

3.7.2 Rancangan *Interface* Mudharabah

Pada rancangan *Interface* mudharabah terdapat 2 jenis tampilan, yaitu berupa *Interface* untuk mengisi dan menampilkan data yang ingin dihitung dan *Interface* menu *save*. Rancangan menu mudharabah ditampilkan pada Gambar 3.13 dan 3.14.



Gambar 3.13 Rancangan *Interface* Mudharabah

Gambar 3.14 Rancangan *Interface* Menu Save

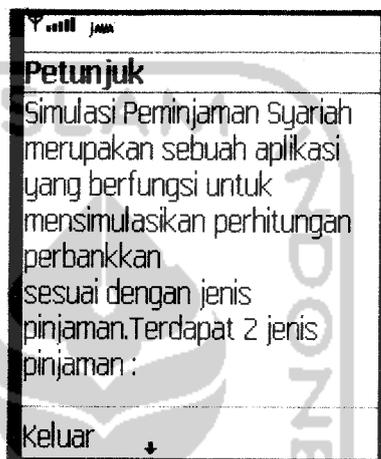
3.7.3 Rancangan *Interface* Murabahah

Interface Murabahah berfungsi untuk menampilkan input data. Selain itu juga untuk menampilkan hasil output data berupa administrasi dan angsuran bulan ini. Rancangan *Interface* murabahah ditampilkan pada Gambar 3.15.

Gambar 3.15 Rancangan Menu Murabahah

3.7.4 Rancangan *Interface* Petunjuk

Interface Petunjuk berfungsi untuk menampilkan informasi penggunaan dari mudharabah dan murabahah. Rancangan *Interface* petunjuk ditampilkan pada Gambar 3.16.



Gambar 3.16 Rancangan *Interface* Petunjuk

BAB IV

PENGUJIAN, ANALISA DAN PEMBAHASAN

4.1 Pengujian Aplikasi

Pengujian sistem merupakan tahapan yang bertujuan untuk memastikan apakah perangkat lunak yang dibuat dapat bekerja secara efektif dan efisien sesuai dengan apa yang diharapkan. Maka dari itu diperlukan pengujian terhadap sistem yang akan diterapkan.

Sebelum aplikasi ini diterapkan dan diimplementasikan, terlebih dahulu perangkat lunak harus memenuhi syarat *error free* (bebas dari kesalahan apapun). Kesalahan dapat terjadi akibat dari kesalahan penulisan bahasa (*script*), kesalahan sewaktu proses atau kesalahan logika. Kesalahan itu akan muncul ketika sedang membangun (*build*) program sebelum dijalankan (*run*) atau juga ketika akan mengaplikasikannya ke *handphone* yang diinginkan tetapi tidak didukung oleh aplikasi itu sendiri. Setelah aplikasi bebas dari kesalahan maka dapat dilakukan pengujian program dengan cara mengaplikasikannya pada *handphone* dan menggunakannya.

4.1.1 Pengujian Umum Aplikasi

Dalam proses pengembangan aplikasi *mobile* informasi pinjaman syariah dengan Java 2 Micro Edition (J2ME) ini banyak ditemukan hal – hal baru yang lebih beragam dan sangat berbeda dari batasan – batasan awal yang telah diterapkan, hal – hal tersebut antara lain :

1. Diperlukannya tampilan menu yang *user friendly* agar *user* lebih mudah untuk mengaksesnya dan dapat menghindari kerancuan dalam penggunaan aplikasi.
2. Jumlah batasan uang yang dimasukkan dalam pinjaman modal dan disesuaikan dengan karakter maksimal yang ditampilkan oleh *handphone*.
3. Batasan Jangka waktu yang ditetapkan sesuai dengan standar yang telah ditentukan.
4. Batasan nisbah atau margin yang telah ditetapkan sesuai dengan standar yang telah ditentukan.

4.1.2 Tahapan Pengujian

Dalam pembuatan aplikasi *mobile* informasi pinjaman syariah dengan Java 2 Micro Edition (J2ME) melalui beberapa penentuan pengujian, antara lain :

1. Penentuan jenis pinjaman

Untuk memberikan batasan pada program, maka dipilih hanya ada dua jenis pinjaman, yaitu : Mudharabah dan Murabahah. Pemilihan pinjaman berdasarkan frekuensi penggunaan *user* dalam melakukan pinjaman.

2. Penetapan batasan jangka waktu dan nisbah atau margin

Jangka waktu setiap pinjaman ditetapkan oleh peraturan dari bank. Sedangkan nisbah atau margin ditetapkan oleh standarisasi pada umumnya.

4.2 Pengujian Sistem Melalui *Handphone*

Prosedur pada tahap pengujian sistem dibagi menjadi beberapa bagian, yaitu

:

1. Prosedur pemilihan menu adalah menu yang menampilkan beberapa pilihan pinjaman dan petunjuk yang disediakan oleh aplikasi *handphone*, berupa : Mudharabah, Murabahah, dan Petunjuk.
2. Prosedur perhitungan biaya angsuran tiap bulan yang langsung dikalkulasikan setelah memasukkan semua input data.

4.2.1 Pengujian Pinjaman Mudharabah

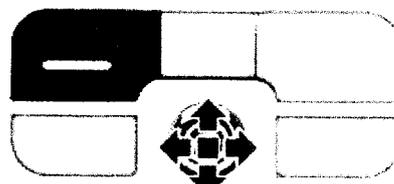
Pengujian pertama adalah melakukan perhitungan pinjaman mudharabah. Sebagai contoh, akan dibuat sebuah kasus pinjaman seorang nasabah ingin meminjam uang untuk menambah modal usaha sebesar Rp. 250.000.000,- dan sanggup melunasi dalam jangka waktu 24 bulan. Setelah dilakukan peninjauan dari pihak bank terhadap nasabah tersebut maka bank menetapkan nisbah sebesar 21,32% dan biaya administarsi sebesar 1% dari uang pinjaman. Angsuran harus dibayar setiap bulan sesuai dengan penghasilan orang tersebut setiap bulan dan nisbah yang telah ditentukan. Maka dari itu kita akan menentukan berapa besar angsuran bulan ini jika penghasilan nasabah tersebut bulan ini adalah Rp. 3.250.000,-. Pengujian akan diperlihatkan oleh beberapa gambar. Gambar 4.1 akan menunjukkan cara pengisian *form*.

Field	Value
Pinjaman Modal (Rp.)	250000000
Jangka Waktu (Bulan)	24
Nisbah (%)	21.32
Administrasi (Rp.)	
Penghasilan Bulan Ini (Rp.)	3250000
Angsuran Bulan Ini (Rp.)	

Gambar 4.1 Cara Pengisian *Form* Mudharabah

Sedangkan hasil perhitungan akan diperlihatkan pada Gambar 4.2.

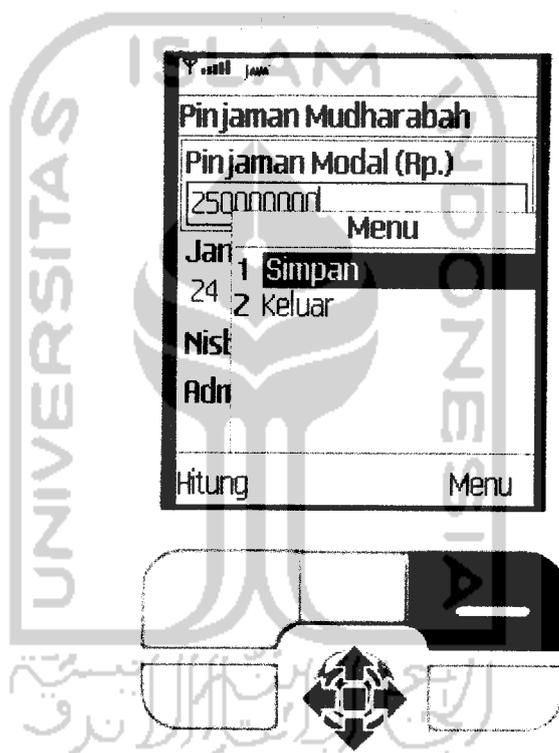
Field	Value
Nisbah (%)	21.32
Administrasi (Rp.)	
Penghasilan Bulan Ini (Rp.)	3250000
Angsuran Bulan Ini (Rp.)	



Gambar 4.2 Hasil Perhitungan Mudharabah

Dilihat dari input yang dimasukkan maka hasil perhitungan biaya angsuran bulan ini sebesar Rp. 11.109.566,- dan biaya administrasi sebesar Rp. 2.500.000,-.

Dalam mudharabah juga terdapat sebuah fungsi simpan yang dapat digunakan untuk menyimpan input yang sudah dimasukkan. Fungsi itu akan berguna ketika kita menggunakan lagi aplikasi mudharabah setelah kita keluar dari aplikasi tersebut. Input akan secara otomatis langsung terisi data input terakhir yang telah dimasukkan sebelumnya. Fungsi simpan terlihat pada Gambar 4.3.



Gambar 4.3 Fungsi Simpan

4.2.2 Pengujian Pinjaman Murabahah

Pengujian Selanjutnya adalah melakukan perhitungan pinjaman murabahah. Sebagai contoh, akan dibuat sebuah kasus pinjaman seorang nasabah ingin meminjam uang untuk membeli rumah. Rumah tersebut seharga Rp. 182.000.000,- dan sanggup melunasi dalam jangka waktu 60 bulan. Setelah

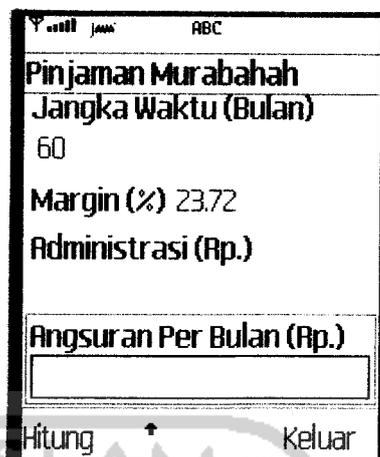
dilakukan peninjauan dari pihak bank terhadap nasabah tersebut maka bank menetapkan margin sebesar 23,72% dan biaya administrasi sebesar 1% dari uang pinjaman. Angsuran harus dibayar setiap bulan sesuai dengan margin yang telah ditentukan. Maka dari itu kita akan menentukan berapa besar angsuran setiap bulannya. Pengujian akan diperlihatkan oleh beberapa gambar. Gambar 4.4 akan menunjukkan cara pengisian form.

Pinjaman Murabahah	
Pinjaman Uang (Rp.)	182000000
Jangka Waktu (Bulan)	60
Margin (%)	23.72
Administrasi (Rp.)	
Hitung	Keluar

Gambar 4.4 Cara Pengisian *Form* Murabahah

Dilihat dari input yang dimasukkan maka hasil perhitungan biaya angsuran setiap bulan adalah Rp. 3.752.840,- dan biaya administrasi sebesar Rp. 1.820.000,-.

Sedangkan hasil perhitungan akan diperlihatkan pada Gambar 4.5.

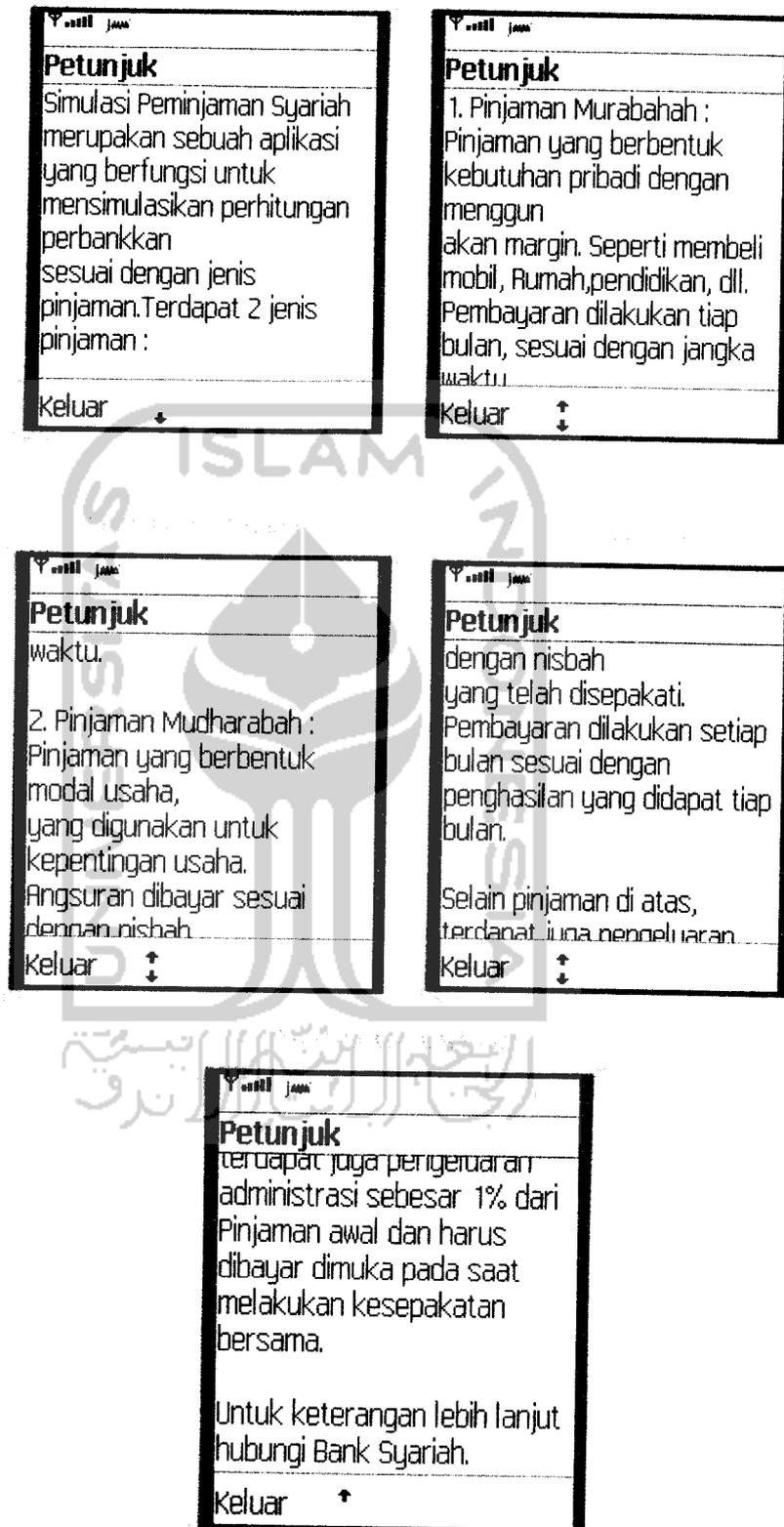


Pinjaman Murabahah
Jangka Waktu (Bulan)
60
Margin (%) 23.72
Administrasi (Rp.)
Angsuran Per Bulan (Rp.)
<input type="text"/>
Hitung ↑ Keluar

Gambar 4.5 Hasil Perhitungan Murabahah

4.2.3 Pengujian Petunjuk

Pada pengujian petunjuk, petunjuk yang digunakan adalah sangat sederhana. Kesederhanaan itu terletak pada fungsi utamanya yang hanya berfungsi untuk menampilkan file teks yang ditulis pada notepad saja. Untuk lebih jelasnya dapat dilihat pada Gambar 4.6.



Gambar 4.6 Petunjuk Syariah

4.3 Analisa Pengujian

Setelah dilakukan pengujian di atas maka perlu dilakukan analisa terhadap aplikasi informasi pinjaman syariah. Dalam analisa pengujian ini akan digunakan 2 (dua) metode yaitu :

1. *Alfa Test*

Pengujian *Alfa test* adalah pengujian sistem yang dilakukan oleh para pemakai sehingga dapat diperoleh tanggapan dari pemakai tentang aplikasi tersebut, baik dari segi format, tampilan maupun tingkat keramahan (kemudahan) aplikasi.

2. *Black Box Test*

Pengujian *Black box test* adalah pengujian sistem yang dilakukan dengan mengamati keluaran dari berbagai masukan. Aplikasi akan dikatakan baik jika keluaran sistem telah sesuai dengan rancangan yang telah dibuat.

Analisa pengujian ini diusahakan untuk melihat sistem dapat berjalan sesuai dengan tujuan awal. Analisa pengujian diperlukan untuk lebih memahami aplikasi yang dibuat dan memberikan informasi lebih akurat.

4.4 Hasil Analisa Menggunakan *Alfa Test*

Metode *Alfa test* yang dipakai adalah dengan cara mendengarkan penilaian dari *user* yang telah mencoba aplikasi ini sehingga akan diketahui kelebihan dan kekurangan program yang dibuat, baik segi format, tingkat kemudahan pengoperasian, maupun input atau output aplikasi.



Kesimpulan dari *alfa test* yang didapat dari *user* yang telah mencoba ternyata memiliki kelebihan dan kekurangan. Kelebihan yang didapat meliputi :

1. Mudah dalam pengoperasian.
2. Mudah dipahami.
3. Perhitungan sangat akurat.

Selain memiliki kelebihan juga terdapat kekurangan, antara lain :

1. Menu pilihan pinjaman kurang banyak.
2. Bukan berupa menu.
3. Tidak mendukung tipe – tipe *handphone* selain Sony Ericsson.

Kelebihan dan kekurangan diatas merupakan pendapat dari *user* yang telah mencoba. Untuk lebih jelasnya mengenai kelebihan dan kekurangan dari aplikasi ini dapat dilihat pada sub bab 4.6.

4.5 Hasil Analisa Menggunakan Metode *Black Box Test*

Dalam analisa menggunakan metode *Black Box Test* pada setiap pengujian akan ditemukan prosedur normal dan tidak normal. Prosedur normal merupakan keberhasilan dari perhitungan yang dilakukan. Sedangkan prosedur tidak normal adalah dimana ketika melakukan perhitungan terdapat kesalahan – kesalahan yang akan ditampilkan jenis kesalahannya.

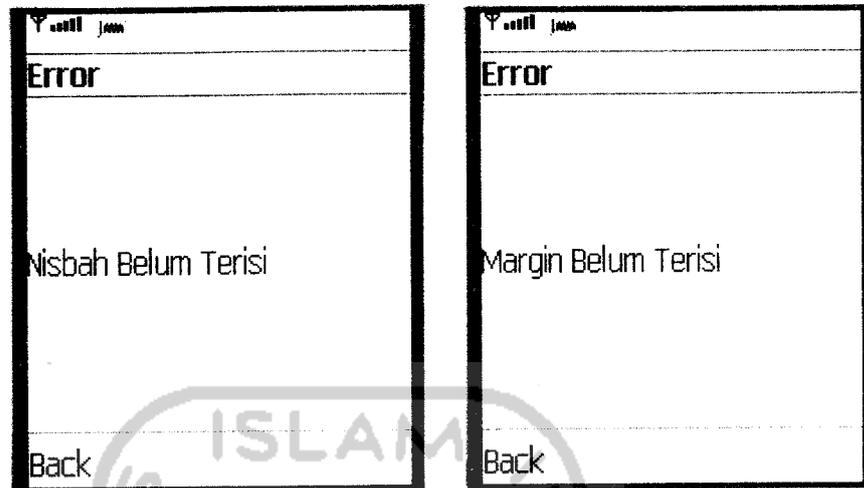
4.5.1 Prosedur Normal Mudharabah dan Murabahah

Perhitungan mudharabah dan Murabahah akan ditampilkan hasilnya jika *user* memasukkan input yang dibutuhkan oleh setiap *form* yang ada.

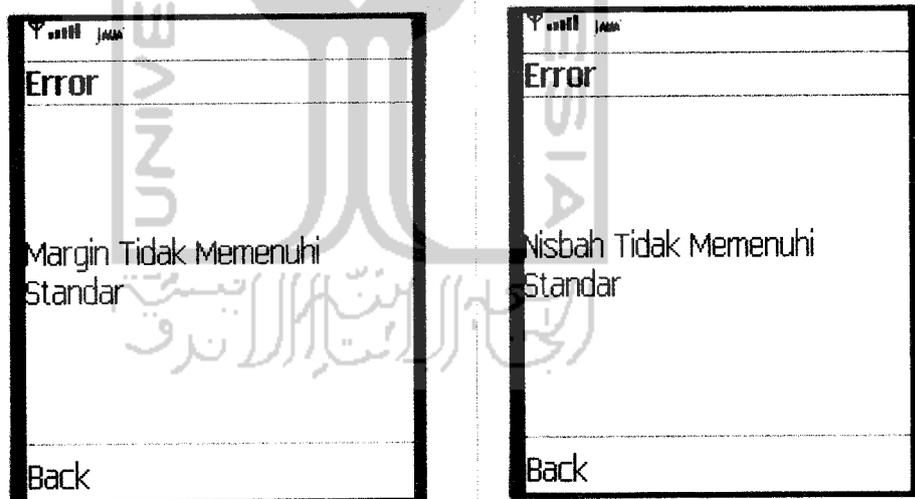
4.5.2 Prosedur Tidak Normal Mudharabah dan Murabahah

Prosedur tidak normal akan terjadi apabila *user* tidak memasukkan input pada setiap *form* yang ada kemudian *user* menekan tombol hitung atau melebihi batas maksimal dari batas yang ditentukan. Jika hal ini terjadi maka pesan kesalahan akan keluar seperti gambar – gambar berikut ini. Untuk Gambar 4.7 merupakan pesan kesalahan yang keluar jika nilai nisbah atau margin belum dimasukkan. Sedangkan Gambar 4.8 menunjukkan pesan kesalahan yang keluar jika dimasukkan nisbah atau margin tidak sesuai dengan standar Begitu juga dengan Gambar 4.9 merupakan pesan kesalahan yang keluar jika pinjaman modal belum terisi. Pada Gambar 4.10 merupakan pesan kesalahan yang keluar jika jangka waktu belum diisi dan juga untuk memberitahukan jika jangka waktu melebihi dari batas yang ditentukan.

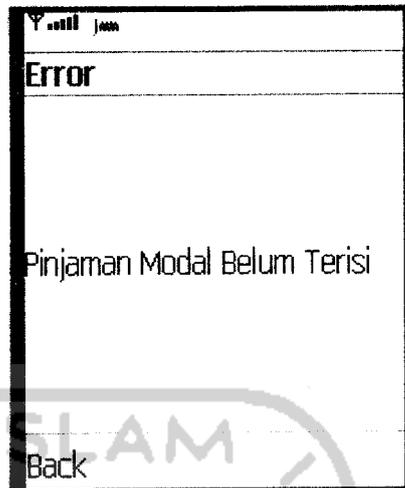
Dari penjelasan pesan kesalahan pada gambar dibawah dapat dijelaskan bahwa pesan kesalahan – kesalahan yang dimiliki oleh kedua pinjaman hampir sama. Hal tersebut dikarenakan input masukan ke duanya hampir sama. Ada juga pesan kesalahan yang memiliki arti yang sama tapi berbeda parameter penentu kesalahan. Salah satunya adalah jangka waktu, pada mudharabah jangka waktu maksimal 60 bulan. Sedangkan untuk murabahah jangka waktunya maksimalnya adalah 120 bulan. Selain itu untuk keterangan pesan kesalahan nisbah dan margin tidak memenuhi standar terjadi apabila bunga melebihi dari angka 100.



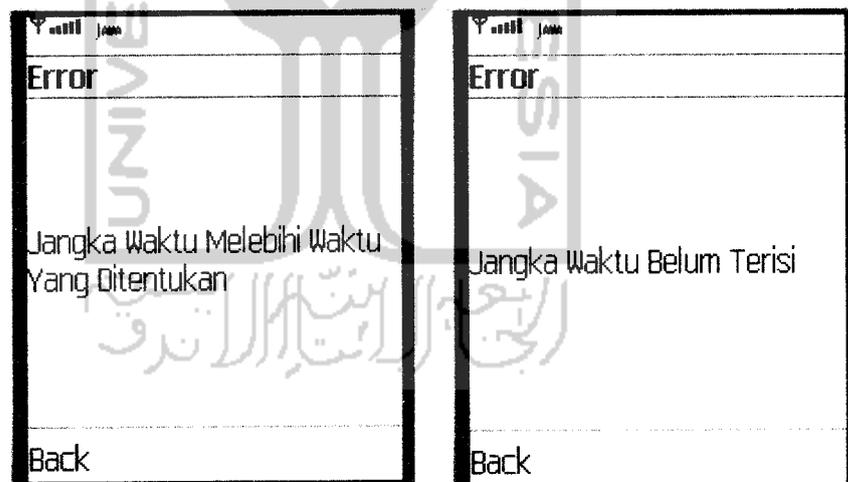
Gambar 4.7 Pesan Kesalahan Input Nisbah dan Margin



Gambar 4.8 Pesan Kesalahan Nisbah dan Margin melebihi Standar



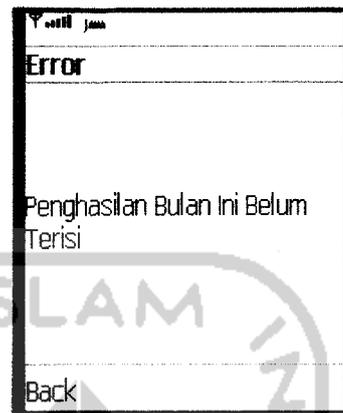
Gambar 4.9 Pesan Kesalahan Pada Input Modal



Gambar 4.10 Pesan Kesalahan Jangka Waktu

Selain terdapat pesan kesalahan yang sama, ada juga pesan kesalahan yang berdiri sendiri. Pada mudharabah terdapat *form* pengisian penghasilan bulan

ini, jika tidak diisi maka pesan tersebut akan muncul. Seperti terlihat pada Gambar 4.11.



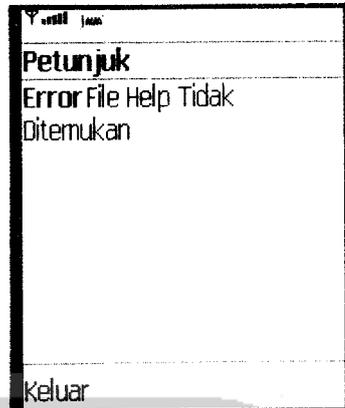
Gambar 4.11 Pesan Kesalahan Penghasilan Bulan Ini

4.5.3 Prosedur Normal Petunjuk

Proses petunjuk hanya menampilkan informasi tentang syariah terutama yang berhubungan dengan pinjaman – pinjaman.

4.5.4 Prosedur Tidak Normal Petunjuk

Prosedur tidak normal terjadi jika *user* tidak bisa membaca file *help*. Kemungkinan ini disebabkan tidak terdapatnya file teks pada direktori *res.* teks tersebut berisi informasi yang berkaitan dengan pinjaman. Untuk lebih jelasnya dapat dilihat pada Gambar 4.12.



Gambar 4.12 Pesan Kesalahan Petunjuk

4.6 Kelebihan dan Kekurangan Program

Setiap aplikasi belum bisa dikatakan sempurna jika belum diketahui kelebihan dan kekurangannya. Untuk itu diperlukan pemahaman mengenai kelebihan dan kekurangan aplikasi informasi pinjaman syariah.

Kelebihan dari pembuatan aplikasi informasi pinjaman syariah adalah :

- a. Mempermudah *user* melakukan perhitungan pinjaman sesuai dengan pinjaman yang dibutuhkan.
- b. *User* bisa menghemat waktu dalam melakukan perhitungan tanpa harus menghitung secara manual.
- c. Tidak perlu datang ke Bank jika hanya ingin mensimulasikan pinjaman.

Sedangkan kekurangan dari pembuatan aplikasi informasi pinjaman syariah adalah :

- a. Fasilitas pilihan pinjaman hanya sebatas dua jenis.
- b. Dikarenakan sifatnya hanya sebatas sebuah simulasi pemodelan maka tidak bisa menjamin bahwa hasil yang didapat sama dengan yang dilakukan di bank.
- c. Tidak semua *handphone* cocok menggunakan aplikasi ini, untuk sementara hanya untuk *handphone* Sony Ericsson saja. Sebab *emulator* yang digunakan adalah milik Sony Ericsson.



BAB V

PENUTUP

5.1 Kesimpulan

Dari Perancangan, pengujian dan analisa yang telah dilakukan ada beberapa kesimpulan yang dapat diambil sebagai hasil dari penelitian, antara lain :

1. Pembuatan aplikasi ini dapat diakses melalui *handphone* yang berbasis java. Terutama yang sudah mendukung MIDP 2.0 dan memiliki konfigurasi CLDC 1.1.
2. Pembuatan aplikasi ini berfungsi untuk mensimulasikan perhitungan – perhitungan perbankan terutama berhubungan dengan perhitungan pinjaman syariah.
3. Terdapat pemilihan menu di dalam aplikasi ini, menu pilihan tersebut berupa *class* (Mudharabah, Murabahah, Petunjuk) dari turunan *package* syariah.
4. Pada perhitungan mudharabah terdapat fungsi simpan (*save*) yang berfungsi untuk menyimpan parameter – parameter input yang ada. Dengan demikian setelah keluar dan masuk lagi *user* tinggal memasukkan input penghasilan bulan ini. Sebab parameter yang dimasukkan tadi langsung secara otomatis keluar dan menempati masing – masing parameter.

5.2 Saran

Disadari bahwa aplikasi ini masih banyak kekurangan dan kelemahan, oleh karena itu untuk yang ingin mengembangkan aplikasi ini selanjutnya disarankan :

1. Membuat menu yang lebih menarik dan dapat melakukan fungsi *back*.
2. Aplikasi perlu dikembangkan agar bisa terkoneksi melalui web dengan database bank syariah.
3. Perlu ditambahkan *history* berapa jumlah pinjaman yang dibayar dan kekurangannya.



DAFTAR PUSTAKA

- Bambang, H., 2003. *Esensi – Esensi Pemrograman Java*. Bandung : Informatika Bandung.
- Antonius, A. H., 2004. *Modul Pelatihan J2ME dan Pengembangan platform Aplikasi Mobile Masa Depan*. Yogyakarta : Himpunan Mahasiswa Elektronika Universitas Negeri Yogyakarta.
- Suyoto, DR., 2005. *Membuat Sendiri Aplikasi Ponsel*. Yogyakarta : Gava Media.
- Shalahuddin, M. dan Rosa, A.S, 2006. *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung : Penerbit Informatika.
- Setianto, D., 2006. *Panduan Lengkap Pemrograman Celluler Device dengan J2ME*. Yogyakarta : Ardana Media.
- Antonius, A. H., 2004. *Pemrograman Mobile Java dengan MIDP.2.0*. Yogyakarta : Penerbit Andi.
- <http://forum.chip.co.id/>



```

package mudharabah;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.rms.*;
import java.util.*;
import mudharabah.Preferences;
public class Mudharabah extends MIDlet implements CommandListener {

    private static final int NUM_SIZE=11;
    private static String pinjaman_modal="10000";
    private static String jangka_waktu="10";
    private static String nisbah="10.0";
    private Preferences mPreferences;
    private TextField t1,t2,t3;
    private Form f;

    private Command saveCmd = new Command("Simpan", Command.SCREEN,
7);

    private final Command exitCmd = new Command("Keluar",
Command.EXIT, 2);

    private final Command calcCmd = new Command("Hitung",
Command.SCREEN, 1);

    private final TextField t4 = new TextField("Penghasilan Bulan Ini
(Rp.)", "", NUM_SIZE,
TextField.DECIMAL);

    private final TextField tr = new TextField("Administrasi (Rp.)",
"", NUM_SIZE,
TextField.UNEDITABLE);

    private final TextField tr1 = new TextField("Angsuran Bulan Ini
(Rp.)", "", NUM_SIZE,
TextField.UNEDITABLE);

    private final Alert alert = new Alert("Error", "", null,
AlertType.ERROR);

    private boolean isInitialized = false;
    public Mudharabah(){}

    protected void startApp() {
        if (isInitialized) {
            return;
        }

        try {
            mPreferences = new Preferences("preferences");
        }
        catch (RecordStoreException rse) {
            Form f = new Form("Exception");
            f.append(new StringItem(null, rse.toString()));
            f.addCommand(new Command("Exit", Command.EXIT, 0));
            f.setCommandListener(this);
            return;
        }

        Form f = new Form("Pinjaman Mudharabah");
        t1 = new TextField("Pinjaman Modal (Rp.)", mPreferences.get
(pinjaman_modal), NUM_SIZE,
TextField.DECIMAL);
        t2 = new TextField("Jangka Waktu (Bulan)", mPreferences.get

```

```

(jangka_waktu), 3,
    TextField.DECIMAL);
    t3 = new TextField("Nisbah (%)", mPreferences.get(nisbah), 5,
        TextField.DECIMAL);
    f.append(t1);
    f.append(t2);
    f.append(t3);
    f.append(tr);
    f.append(t4);
    f.append(tr1);
    f.addCommand(exitCmd);
    f.addCommand(calcCmd);
    f.addCommand(saveCmd);
    f.setCommandListener(this);
    Display.getDisplay(this).setCurrent(f);
    alert.addCommand(new Command("Back", Command.SCREEN, 1));
    isInitialized = true;

}

protected void destroyApp(boolean unconditional) {}

protected void pauseApp() {}

public void commandAction(Command c, Displayable d) {
    if (c == exitCmd) {
        destroyApp(false);
        notifyDestroyed();
        return;
    }
    else if (c == saveCmd) {
        mPreferences.put(pinjaman_modal, t1.getString());
        mPreferences.put(jangka_waktu, t2.getString());
        mPreferences.put(nisbah, t3.getString());
        try { mPreferences.save(); }
        catch (RecordStoreException rse) {}
    }
    else if (c == calcCmd) {
        long res = 0;
        double res1 = 0.0;

        try {
            long n1 = getNumberLong(t1, "Pinjaman Modal");
            int n2 = getNumberInt(t2, "Jangka Waktu");
            if (n2 > 60)
            {
                alert.setString("Jangka Waktu Melebihi Waktu Yang
Ditentukan");
                Display.getDisplay(this).setCurrent(alert);
                throw new NumberFormatException();
            }
            else
            {
                double n3 = getNumber(t3, "Nisbah");
                if (n3 > 100)
                {
                    alert.setString("Nisbah Tidak Memenuhi Standar");
                    Display.getDisplay(this).setCurrent(alert);
                    throw new NumberFormatException();
                }
                else
                {
                    long n4 = getNumberLong(t4, "Penghasilan Bulan Ini");
                    res= n1/100;
                    res1= (n4*n3/100)+(n1/n2);
                }
            }
        }
    }
}

```

```

    }
}
} catch (NumberFormatException e) {
    return;
} catch (ArithmeticException e) {
    alert.setString("Pembagi Tidak Boleh Nol");
    Display.getDisplay(this).setCurrent(alert);
    return;
}

String res_str = Long.toString(res);
Double angka = new Double(res1);
long res1_long=angka.longValue();
String res1_str=Long.toString(res1_long);

    if (res_str.length() > tr.getMaxSize()) {
        tr.setMaxSize(res_str.length());
    }
    tr.setString(res_str);
    if (res1_str.length() > tr1.getMaxSize()) {
        tr1.setMaxSize(res1_str.length());
    }
    tr1.setString(res1_str);
}
}

private int getNumberInt(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    int n;

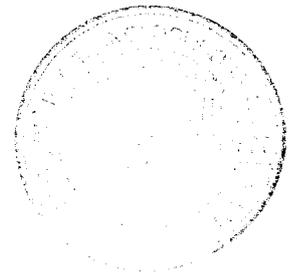
    try {
        n = Integer.parseInt(s);
    } catch (NumberFormatException e) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}

private long getNumberLong(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    long n;

    try {
        n = Long.parseLong(s);
    } catch (NumberFormatException e) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}

```



```

}
private double getNumber(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    double n;

    try {
        n = Double.parseDouble(s);
    } catch (NumberFormatException e) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}
}

```



```

package mudharabah;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.rms.*;
import java.util.*;

public class Preferences {
    private String mRecordStoreName;
    private Hashtable mHashtable;
    public Preferences(String recordStoreName)
        throws RecordStoreException {
        mRecordStoreName = recordStoreName;
        mHashtable = new Hashtable();
        load();
    }
    public String get(String key) {
        return (String)mHashtable.get(key);
    }
    public void put(String key, String value) {
        if (value == null) value = "";
        mHashtable.put(key, value);
    }
    private void load() throws RecordStoreException {
        RecordStore rs = null;
        RecordEnumeration re = null;

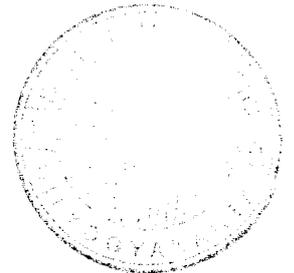
        try {
            rs = RecordStore.openRecordStore(mRecordStoreName, true);
            re = rs.enumerateRecords(null, null, false);
            while (re.hasNextElement()) {
                byte[] raw = re.nextRecord();
                String pref = new String(raw);
                // Parse out the name.
                int index = pref.indexOf('|');
                String name = pref.substring(0, index);
                String value = pref.substring(index + 1);
                put(name, value);
            }
        }
        finally {
            if (re != null) re.destroy();
            if (rs != null) rs.closeRecordStore();
        }
    }

    public void save() throws RecordStoreException {
        RecordStore rs = null;
        RecordEnumeration re = null;
        try {
            rs = RecordStore.openRecordStore(mRecordStoreName, true);
            re = rs.enumerateRecords(null, null, false);

            while (re.hasNextElement()) {
                int id = re.nextRecordId();
                rs.deleteRecord(id);
            }

            Enumeration keys = mHashtable.keys();
            while (keys.hasMoreElements()) {
                String key = (String)keys.nextElement();
                String value = get(key);
                String pref = key + "|" + value;
                byte[] raw = pref.getBytes();
                rs.addRecord(raw, 0, raw.length);
            }
        }
    }
}

```



```
finally {  
    if (re != null) re.destroy();  
    if (rs != null) rs.closeRecordStore();  
}  
}  
}
```



```

package murabahah;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.*;
import java.util.*;

public class Murabahah extends MIDlet implements CommandListener {

    private static final int NUM_SIZE=11;

    private final Command exitCmd = new Command("Keluar",
Command.EXIT, 2);

    private final Command calcCmd = new Command("Hitung",
Command.SCREEN, 1);

    private final TextField t1 = new TextField("Pinjaman Uang (Rp.)",
"", NUM_SIZE,
        TextField.DECIMAL);

    private final TextField t2 = new TextField("Jangka Waktu
(Bulan)", "", 3,
        TextField.DECIMAL);

    private final TextField t3 = new TextField("Margin (%)", "", 5,
        TextField.DECIMAL);

    private final TextField tr = new TextField("Administrasi (Rp.)",
"", NUM_SIZE,
        TextField.UNEDITABLE);

    private final TextField tr1 = new TextField("Angsuran Per Bulan
(Rp.)", "", NUM_SIZE,
        TextField.UNEDITABLE);

    private final Alert alert = new Alert("Error", "", null,
AlertType.ERROR);

    private boolean isInitialized = false;

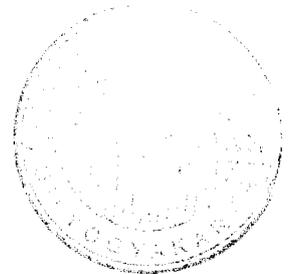
    protected void startApp() {
        if (isInitialized) {
            return;
        }
        Form f = new Form("Pinjaman Murabahah");
        f.append(t1);
        f.append(t2);
        f.append(t3);
        f.append(tr);
        f.append(tr1);
        f.addCommand(exitCmd);
        f.addCommand(calcCmd);
        f.setCommandListener(this);
        Display.getDisplay(this).setCurrent(f);
        alert.addCommand(new Command("Back", Command.SCREEN, 1));
        isInitialized = true;
    }

    protected void destroyApp(boolean unconditional) {}

    protected void pauseApp() {}

    public void commandAction(Command c, Displayable d) {
        if (c == exitCmd) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}

```



```

        return;
    }
    long res = 0;
    double res1 = 0.0;

    try {
        long n1 = getNumberLong(t1, "Pinjaman Uang");
        int n2 = getNumberInt(t2, "Jangka Waktu");
        if (n2 > 120)
        {
            alert.setString("Jangka Waktu Melebihi Waktu Yang
Ditentukan");
            Display.getDisplay(this).setCurrent(alert);
            throw new NumberFormatException();
        }else
        {
            double n3 = getNumber(t3, "Margin");
            if (n3 > 100)
            {
                alert.setString("Margin Tidak Memenuhi Standar");
                Display.getDisplay(this).setCurrent(alert);
                throw new NumberFormatException();
            }else
            {
                res= n1/100;
                res1= ((n3/100*n1)+n1)/n2;
            }
        }

    } catch (NumberFormatException e) {
        return;
    } catch (ArithmeticException e) {
        alert.setString("Divide by zero.");
        Display.getDisplay(this).setCurrent(alert);
        return;
    }

    String res_str = Long.toString(res);
    Double angka = new Double(res1);
    long res1_long=angka.longValue();
    String res1_str=Long.toString(res1_long);

    if (res_str.length() > tr.getMaxSize()) {
        tr.setMaxSize(res_str.length());
    }
    tr.setString(res_str);
    if (res1_str.length() > tr1.getMaxSize()) {
        tr1.setMaxSize(res1_str.length());
    }
    tr1.setString(res1_str);
}

private int getNumberInt(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    int n;

    try {
        n = Integer.parseInt(s);
    } catch (NumberFormatException e) {

```

```

        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}

private long getNumberLong(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    long n;

    try {
        n = Long.parseLong(s);
    } catch (NumberFormatException e) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}

private double getNumber(TextField t, String type)
    throws NumberFormatException {
    String s = t.getString();

    if (s.length() == 0) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw new NumberFormatException();
    }
    double n;

    try {
        n = Double.parseDouble(s);
    } catch (NumberFormatException e) {
        alert.setString(type + " Belum Terisi");
        Display.getDisplay(this).setCurrent(alert);
        throw e;
    }
    return n;
}
}

```



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

public class Help
    extends MIDlet {

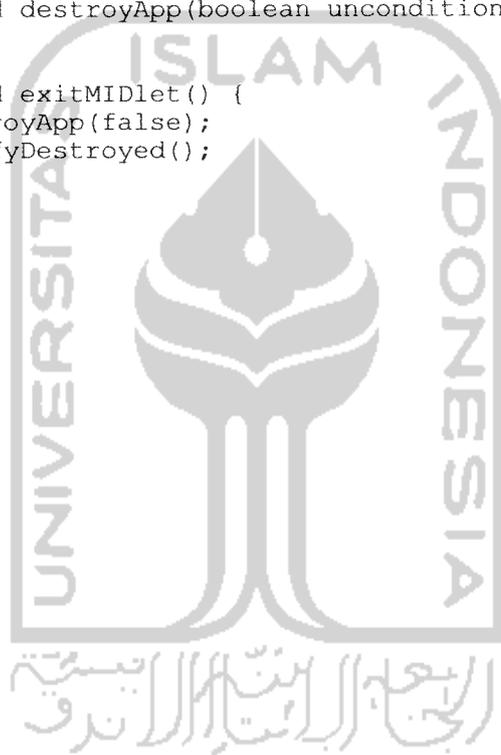
    Display display;
    ReadFile read;

    public Help() {
        display = Display.getDisplay(this);
    }
    public void startApp() {
        read = new ReadFile(this, display);
        display.setCurrent(read);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {
    }

    public void exitMIDlet() {
        destroyApp(false);
        notifyDestroyed();
    }
}
```



```

import javax.microedition.lcdui.*;
import java.io.*;

public class ReadFile
    extends Form
    implements CommandListener {
    private Help midlet;
    private Display display;
    private Command cmExit;
    private String help;
    private StringItem si;

    public ReadFile(Help midlet, Display display) {
        super("Petunjuk");
        this.midlet = midlet;
        this.display = display;

        if ((help = readHelpText()) != null) {
            si = new StringItem("", help);
        } else
            si = new StringItem("Error", "File Help Tidak
Ditemukan");

        append(si);

        cmExit = new Command("Keluar", Command.EXIT, 1);
        setCommandListener(this);
        addCommand(cmExit);
    }

    public void commandAction(Command c, Displayable s) {
        if (c == cmExit)
            midlet.exitMIDlet();
    }

    private String readHelpText() {
        InputStream is = getClass().getResourceAsStream
("help.txt");
        try {
            StringBuffer sb = new StringBuffer();
            int chr, i = 0;
            while ((chr = is.read()) != -1)
                sb.append((char) chr);
            return sb.toString();
        } catch (Exception e) {
        }
        return null;
    }
}

```