

SKRIPSI

DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA



Disusun Oleh :

Nama : HERRU PRASETIYA
NIM : 0412690



**KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2008**

LEMBAR PERSETUJUAN

DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)*

Disusun Oleh :

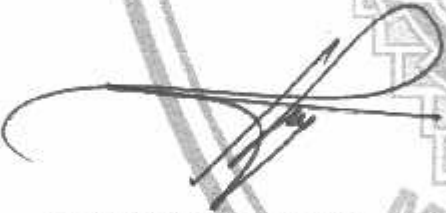
HERRU PRASETIYA

NIM : 04.12.690

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Ir. F. Yudi Limpraptono, MT
NIP Y. 1039500274


M. Ashar, ST, MT
NIP. P 103 0500 408



Mengetahui

Ketua Jurusan Teknik Elektro S-1


Ir. F. Yudi Limpraptono, MT
NIP Y. 1039500274

**KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA S-1
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2008**



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : Herru Prasetya
NIM : 0412690
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : Desain Aplikasi Mobile Banking Menggunakan JAVA

Dipertahankan dihadapan tim penguji skripsi jenjang Strata Satu (S-1) pada:

Hari : Jumat

Tanggal : 26 September 2008

Dengan Nilai : 88.75 (A) *84*

PANITIA UJIAN SKRIPSI



Ir. Mochtar Asroni, MSME
NIP.Y. 1018100036

SEKRETARIS

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

ANGGOTA PENGUJI

PENGUJI I

Dr. Cahyo Crvysdian
NIP. Y. 103040412

PENGUJI II

Sotyojadi, ST, MT
NIP.Y. 1039700309

DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA

HERRU PRASETIYA

04.12.690

**Jurusan Teknik Elektro, Konsentrasi Teknik Komputer & Informatika,
Fakultas Teknologi Industri, Institut Teknologi Nasional, Jl. Karanglo KM.2,
Tasik Madu Malang, Ruhe_Blaz@yahoo.com,
ruhe_developer@dev.motorola.com**

Dosen Pembimbing : I. Ir.F.Yudi Limpraptono, MT.

II. M.Ashar, ST, MT

**Kata Kunci : e- banking, sms banking, internet, mirror server, keylogger, Java,
GPRS**

Perbankan saat ini mengaplikasikan teknologi informasi untuk memenuhi layanan nasabah, teknologi yang di gunakan saat ini adalah e-banking & sms banking. keduanya memiliki kelemahan dan kelebihan, jika e-banking untuk melakukan transaksi di butuhkan koneksi internet dan bahaya mirror server, keylogger dan ini tidak semua lapisan bisa menikmati, sedangkan jika sms banking sangat mudah dilakukan tetapi ada banyak format yang harus dihafal untuk melakukan transaksi, oleh karena itu di buat suatu aplikasi mobile banking berbasis JAVA dan berkomunikasi dengan server menggunakan jaringan GPRS yang menggabungkan antara kelebihan e-banking & SMS banking, dimana user bisa melakukan transaksi perbankan dimana saja dan kapan saja menggunakan handphone.

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA” ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer & Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
2. Bapak Ir. F. Yudi Limpraptono, MT selaku Dosen Pembimbing I
3. Bapak M.Ashar, ST, MT selaku Dosen Pembimbing II dan Ka. Laboratorium Pemrograman Komputer & Multimedia.
4. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
5. Rekan-rekan Instruktur di Laboratorium Pemrograman Komputer & Multimedia.
6. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu

penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, September 2008

Penyusun

LEMBAR PERSEMBAHAN

Satu episode kehidupan sudah aku lalui, sebuah episode yang mengenalkan ku pada arti hidup, dimana hidup harus berjuang meski dimulai dari NOL, hidup yang lebih menghargai, hidup yang lebih berarti dengan prestasi, semua yang kuraih tak lepas dari dukungan banyak pihak yang mungkin aku tak kan bisa membalas jasa dan pengorbanan mereka selama ini, mereka telah menemani setiap langkahku, hanya ini sekarang yang bisa ku persembahkan pada mereka, suatu persembahan yang sangat berarti.

1. Kedua orang tuaku & Keluarga, terima kasih atas dukungan moral & financial yang telah support aku selama ini, semua itu sangat berarti.
 2. Bapak ASHAR,MT,ST , terima kasih pak atas semuanya, bapak bukan sekedar dosen, tapi juga sahabat yang terbaik selalu mendukung tiap langkah, dan selalu percaya bahwa saya bisa, pencapaian prestasi saya & teman2 selama ini mungkin belum bisa dibanggakan tapi saya berjanji untuk tetap berkarya demi Ilmu Pengetahuan demi cita – cita kita, terima kasih telah menuntun saya ke dunia JAVA, saya mendapatkan banyak hal yang tak terduga.
 3. TEAM, (Agit, Sahru, Abdi) Bro... tanpa kalian aku mungkin bukan apa – apa kalian sangat berperan dalam kehidupan ku beberapa tahun ini kalian mengajarkan banyak hal (tanpa sadar), Agit yang cerdas, Sahru yang inovatif, Abdi yang Telaten & sabar, aku berharap TEAM kita adalah TEAM yang solid dan bisa mewujudkan cita-cita kita, perjuangan belum berakhir, ini adalah awal perjuangan sesungguhnya
がんばって ください !!
 4. Perempuanaku, EVA.. Terima kasih sudah sabar menemaniku di awal dan akhir episode ini, memberikan inspirasi dan tujuan hidup, aku berharap perempuanku juga temani aku di episode selanjutnya amin.
 5. Sahabat LAB Pemrograman Komputer & Multimedia angkatan 04,05,06,07 Thank bro !!
 6. Semua sahabat Teknik Informatika ITN ありがとう !!
-

7. Seluruh civitas Dosen & Karyawan ITN Malang.
8. Keluarga besar di Malang (rumah kedua).

SECOND WORLD



DAFTAR ISI

LEMBAR PERSETUJUAN	i
BERITA ACARA.....	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1. Bank	5
2.2. JAVA	8
2.2.1. J2ME	9
2.2.1.1. <i>Configuration</i>	11
2.2.1.2. <i>Profile</i>	11
2.2.1.3. <i>CLDC</i>	12
2.2.1.4. <i>Generic Connection Framework</i>	12

2.2.1.5. MIDP	13
2.2.1.6. Midlet	14
2.2.1.7. Siklus Midlet	15
2.3. Web Server Apache	15
2.4. PHP	17
2.5. WAP	18
2.6. GPRS	20
 BAB III DESAIN DAN ANALISA SISTEM MOBILE BANKING	24
3.1. Spesifikasi Sistem	24
3.1.1. Spesifikasi Software	25
3.1.2. Spesifikasi Hardware	27
3.2. Desain Keseluruhan Sistem	28
3.2.1. Desain Komunikasi Http	29
3.2.2. Desain Transaksi	34
3.2.3. Desain Konten	38
3.2.4. Desain Server	40
3.2.5. Desain Database	41
 BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	
MOBILE BANKING	43
4.1. Pengujian Koneksi IHTTP	43
4.2. Pengujian Login	48
4.3. Pengujian Isi Pulsa	49

4.4. Pengujian Transfer	50
4.5. Pengujian Cek Saldo	52
4.6. Pengujian Informasi	52
4.7. Pengujian Ganti Pin	53
4.8. Pengujian Pembayaran Tagihan	54
4.9. Pengujian Logout	55
4.10. Pengujian Tarif & Data GPRS IM3	56
4.11. Pengujian Kinerja Aplikasi	56
 BAB V PENUTUP	 58
5.1. Kesimpulan	58
5.2. Saran	58
 DAFTAR PUSTAKA	 59
 LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

2-1. Platform JAVA	9
2-2. Arsitektur J2ME	10
2-3. Koncksi GCF (Generic Connection Framework)	13
2-4. Siklus Hidup Midlet	15
2-5. Layer WAP	18
2-6. Cara Kerja WAP	19
2-7. Cara Kerja GPRS	22
3-1. Emulator e-398	27
3-2. Motorola c-398	27
3-3. Desain Keseluruhan Sistem	28
3-4. Analisa Komunikasi HTTP secara Hardware	29
3-5. Flowchart Komunikasi Http	32
3-6. Pertukaran Key oleh Server	35
3-7. Flowchart transaksi	37
3-8. Desain Konten Mobile Banking	39
3-9. Desain servlet server	40
3-10. Tbl_data_user	41
3-11. Tbl_halo	41
3-12. Tbl_Matrix	41
3-13. Tbl_Pln	42
3-14. Tbl_Speedy	42
3-15. Tbl_Telkom	42

4-1. Arsitektur <i>Low – End Consumer Devices</i>	44
4-2. Hirarki GCF	44
4-3. Layer WAP	45
4-4. Hasil Komunikasi Http	47
4-5. Login & Periksa Data.....	49
4-6. . Login Sukses & Gagal	49
4-7. Pengisian pulsa & Periksa Data	50
4-8. Konfirmasi Pengisian Pulsa	50
4-9. Saldo Mencukupi & Saldo Tidak Mencukupi (isi Pulsa)	50
4-10. Transfer & Konfirmasi	51
4-11. Saldo Mencukupi & Saldo Tidak Mencukupi (Transfer).....	51
4-12. Cek Saldo	52
4-13. Informasi dari Server	52
4-14. Pergantian PIN & Konfirmasi Data	53
4-15. PIN Salah & PIN telah di ubah	53
4-16. Pembayaran & Konfirmasi	54
4-17. Nomer Pelanggan Salah & Data Pelanggan	55
4-18. Hasil Pegecekan Status	55
4-19. Saldo Mencukupi & Saldo Tidak Mencukupi (Tagihan)	55
4-20. Konfirmasi Logout & Logout	56

DAFTAR TABEL

2-1. Platform Java	9
3-1. Response code Koneksi HTTP.....	33
4-1. Pengujian Waktu & Tarif.....	56
4-2. Kinerja Aplikasi	57

BAB I

PENDAHULUAN

1. Pendahuluan

Perkembangan dunia teknologi komputer saat ini sangat pesat perkembangannya dan sangat luas aplikasinya, tidak hanya di monopoli pada saat awal perkembangannya yaitu oleh pihak militer saja tapi sudah merambah hingga dunia pendidikan, bahkan untuk saat ini saja setiap badan usaha baik itu swasta ataupun negeri memanfaatkan teknologi computer demi terciptanya kinerja yang lebih baik, begitu pun di dunia perbankan, perkembangan perekonomian dunia yang saat ini maju begitu pesat membuat kebutuhan akan transaksi perbankan di butuhkan secara real time tiap saat demi lancarnya kegiatan perekonomian.

1.1. Latar Belakang

Dunia perbankan saat ini semakin cepat mengaplikasikan berbagai macam teknologi demi terciptanya Sistem yang kuat, dan cepat dalam melayani berbagai kebutuhan nasabah, banyak hal yang ditempuh oleh banyak perbankan untuk hal tersebut, antara lain yaitu dengan hadirnya e-banking yang mana kita bisa melakukan transaksi kapan dan dimana saja, perkembangan e-banking sendiri cukup pesat, pihak perbankan bahkan mengaplikasikan teknologi pengaman terbaru demi keabsahan transaksi yang dilakukan, secara umum ada 2 teknologi yang menjadi basis interaksi dunia perbankan agar dapat melakukan transaksi secara online yaitu Teknologi Internet dan Teknologi telepon selular

Dalam Melakukan transaksi perbankan melalui internet memang sangat mudah dilakukan tetapi karena internet merupakan teknologi terbuka, maka rentan terhadap tindak kejahatan yang menyebabkan kegiatan perbankan melalui internet masih berkurang, sebut saja kejadian mirror bank yang dialami oleh BCA tahun 2001, dan pada bank Niaga tahun 2008, menyebabkan nasabah berfikir dua kali untuk memanfaatkan teknologi tersebut selain itu untuk melakukan transaksi membutuhkan jaringan internet dan ini hanya bisa dilakukan oleh lapisan tertentu saja, Dengan menggunakan teknologi telepon selular, menjangkau seluruh lapisan masyarakat, user bisa memanfaatkan teknologi SMS (*Short Message Service*)

untuk melakukan proses transaksi, dimana user jika ingin bertransaksi harus mengirim format khusus sesuai dengan kebutuhan transaksinya yang telah ditetapkan oleh pihak bank dan ini mempersulit transaksi karena user harus berkali-kali mengirimkan SMS hanya untuk sebuah transaksi.

Java adalah sebuah bahasa pemrograman berorientasi pada objek yang sangat fleksibel, dimana kita bisa menjalankan satu program java di berbagai macam platform yang berbeda, java sendiri saat ini terbagi dalam 3 kategori besar, yaitu J2SE (Java 2 Standart Edition) J2EE (Java 2 Enterprise Edition) dan J2ME (Java 2 Micro Edition). Kategori-kategori tersebut memiliki fungsi yang berbeda-beda, yaitu J2SE biasanya digunakan untuk Aplikasi berbasis Desktop, J2EE biasanya digunakan untuk aplikasi berbasis Enterprise (Client - Server), dan J2ME biasanya digunakan untuk aplikasi pada mobile device, sebagai turunan dari J2SE dan merupakan yang paling berkembang pesat saat ini.

Teknologi J2ME saat ini banyak dimanfaatkan untuk membangun aplikasi pada mobile device, karena dukungan dari vendor handphone yang kuat menyebabkan perkembangan teknologi inipun semakin pesat, dengan menggunakan teknologi J2ME. Adapun keunggulannya antara lain memungkinkan berbagai kegiatan perbankan dilakukan oleh user menggunakan handphone yang mendukung J2ME.

Dengan Teknologi J2ME dapat di desain sebuah aplikasi berbasis mobile dimana aplikasi tersebut di desain untuk mempermudah transaksi perbankan oleh user, sehingga user bisa melakukan transaksi perbankan dengan lebih mudah, menggunakan telepon selular.

1.2. Rumusan Masalah

Berdasarkan hal di atas maka timbul suatu perumusan permasalahan yang peneliti ajukan yaitu :

Bagaimana mendesain aplikasi mobile banking menggunakan J2ME yang mendukung transaksi perbankan.

1.3. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah :

Menghasilkan desain aplikasi mobile banking yang mempermudah user dalam melakukan transaksi perbankan.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut :

1. Dalam merancang sistem hanya pada Sistem aplikasi user.
2. Tidak membahas teknologi pengamanan secara luas, pada sisi server bank tetapi hanya pada teknik pengamanan yang di gunakan pada ponsel
3. Tidak membahas macam-macam transaksi secara kompleks.
4. Tidak Membahas Teknologi GPRS secara luas, tetapi hanya pada handphone.

1.5. Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Studi literatur, yaitu tinjauan pustaka untuk mempelajari teori-teori yang terkait dan berhubungan dengan permasalahan melalui media Referensi Buku, dan Literatur dari Internet.
2. Perancangan
Merancang dan mengembangkan desain mobile banking yang sesuai dengan transaksi perbankan umum yang ada di indonesia.
3. Pembuatan Program
Pembuatan Aplikasi Mobile Banking dengan Bahasa Pemrograman JAVA.
4. Implementasi & Pengujian
Merancang dan mengembangkan desain mobile banking berdasarkan data-data yang diperoleh.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Perumusan Masalah, Tujuan dan Manfaat Penelitian, Pembatasan Permasalahan, Metodologi Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan pembahasan yang dilakukan, yang di dalamnya memuat teori-teori tentang Mobile Banking dan JAVA.

Bab III : Desain & Analisis Sistem Mobile Banking

Bab ini membahas Desain dan analisis dari Aplikasi Mobile Banking pada hardware dan software.

Bab IV : Implementasi dan Pengujian Sistem Mobile Banking

Berisi tentang Implementasi dari Hasil Desain Aplikasi pada emulator dan pengujian pada aplikasi.

Bab V : Penutup

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran mengenai hasil yang telah diPeroleh.

BAB II

LANDASAN TEORI

Dalam bab ini akan dibahas mengenai dasar-dasar teori yang mendukung dalam pembuatan Aplikasi Mobile Banking menggunakan JAVA.

2.1. Bank

Bank adalah sebuah tempat di mana uang disimpan dan dipinjamkan. Menurut Undang-undang Negara Republik Indonesia Nomor 10 Tahun 1998 Tanggal 10 November 1998 tentang perbankan, yang dimaksud dengan bank adalah *badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan dan menyalurkannya kepada masyarakat dalam bentuk kredit dan atau bentuk-bentuk lainnya dalam rangka meningkatkan taraf hidup rakyat banyak*. Dari pengertian di atas dapat dijelaskan secara lebih luas lagi bahwa bank merupakan perusahaan yang bergerak dalam bidang keuangan, artinya aktivitas perbankan selalu berkaitan dalam bidang keuangan.

Fungsi utama dari bank adalah menyediakan jasa menyangkut penyimpanan nilai dan perluasan kredit. Evolusi bank berawal dari awal tulisan, dan berlanjut sampai sekarang di mana bank sebagai institusi keuangan yang menyediakan jasa keuangan. Sekarang ini bank adalah institusi yang memegang lisensi bank. Lisensi bank diberikan oleh otoriter supervisi keuangan dan memberikan hak untuk melakukan jasa perbankan dasar, seperti menerima tabungan dan memberikan pinjaman.

Jasa bank sangat penting dalam pembangunan ekonomi suatu negara. Jasa perbankan pada umumnya terbagi atas dua tujuan. Pertama, sebagai penyedia mekanisme dan alat pembayaran yang efisien bagi nasabah. Untuk ini, bank menyediakan uang tunai, tabungan, dan kartu kredit. Ini adalah peran bank yang paling penting dalam kehidupan ekonomi. Tanpa adanya penyediaan alat pembayaran yang efisien ini, maka barang hanya dapat diperdagangkan dengan cara barter yang memakan waktu.

Dalam mendukung segala kegiatan diatas maka seiring dengan perkembangan waktu di butuhkan sebuah Sistem yang lebih canggih dalam melakukan transaksi demi terciptanya kondisi ekonomi yang seimbang adapun berbagai teknologi yang di manfaatkan oleh dunia perbankan antara lain:

1. ***Automated Teller Machine (ATM)***. Terminal elektronik yang disediakan lembaga keuangan atau perusahaan lainnya yang membolehkan nasabah untuk melakukan penarikan tunai dari rekening simpanannya di bank, melakukan setoran, cek saldo, atau pemindahan dana.
2. ***Computer Banking***. Layanan bank yang bisa diakses oleh nasabah melalui koneksi internet ke pusat data bank, untuk melakukan beberapa layanan perbankan, menerima dan membayar tagihan, dan lain-lain.
3. ***Debit (or check) Card***. Kartu yang digunakan pada ATM atau terminal *point-of-sale* (POS) yang memungkinkan pelanggan memperoleh dana yang langsung didebet (diambil) dari rekening banknya
4. ***Direct Deposit*** Salahsatu bentuk pembayaran yang dilakukan oleh organisasi (misalnya pembei kerja atau instansi pemerintah) yang

membayar sejumlah dana (misalnya gaji atau pensiun) melalui transfer elektronik. Dana transfer langsung ke setiap nasabah.

5. ***Electronic Bill Presentment and Payment (EBPP)***. Bentuk pembayaran tagihan yang disampaikan atau diinformasikan ke nasabah atau pelanggan secara online, misalnya melalui email atau catatan dalam rekening bank.
6. ***Electronic Check Conversion***. Proses konversi informasi yang tertuang dalam cek (nomor rekening, jumlah transaksi, dll) ke dalam format elektronik agar bisa dilakukan pemindahan dana elektronik atau proses lebih lanjut
7. ***Electronic Fund Transfer (EFT)***. Perpindahan “uang” atau “pinjaman” dari satu rekening ke rekening lainnya melalui media elektronik
8. ***Payroll Card***. Salah satu tipe “*stored-value card*” yang diterbitkan oleh pemberi kerja sebagai pengganti cek yang memungkinkan pegawainya mengakses pembayarannya pada terminal ATM atau *Point of Sales*.
9. ***Preauthorized Debit (or automatic bill payment)***. Bentuk pembayaran yang mengizinkan nasabah untuk mengotorisasi pembayaran rutin otomatis yang diambil dari rekening banknya pada tanggal-tanggal tertentu dan biasanya dengan jumlah pembayaran tertentu (misalnya pembayaran listrik, tagihan telepon, dll). Dana secara elektronik ditransfer dari rekening pelanggan ke rekening kreditor (misalnya PLN atau PT Telkom).
10. ***Prepaid Card***. Salah satu tipe *Stored-Value Card* yang menyimpan nilai moneter di dalamnya dan sebelumnya pelanggan sudah membayar nilai tersebut ke penerbit kartu.

11. **Smart Card.** Salah satu tipe *stored-value card* yang di dalamnya tertanam satu atau lebih *chips* atau *microprocessors* sehingga bisa menyimpan data, melakukan perhitungan, atau melakukan proses untuk tujuan khusus (misalnya validasi PIN, otorisasi pembelian, verifikasi saldo rekening, dan menyimpan data pribadi).
12. **Stored-Value Card.** Kartu yang di dalamnya tersimpan sejumlah nilai moneter, yang diisi melalui pembayaran sebelumnya oleh pelanggan atau melalui simpanan yang diberikan oleh pemberi kerja atau perusahaan lain. Untuk *single-purpose stored value card*, penerbit (*issuer*) dan penerima (*acceptor*) kartu adalah perusahaan yang sama dan dana pada kartu tersebut menunjukkan pembayaran di muka untuk penggunaan barang dan jasa tertentu (misalnya kartu telpon).

2.2. JAVA

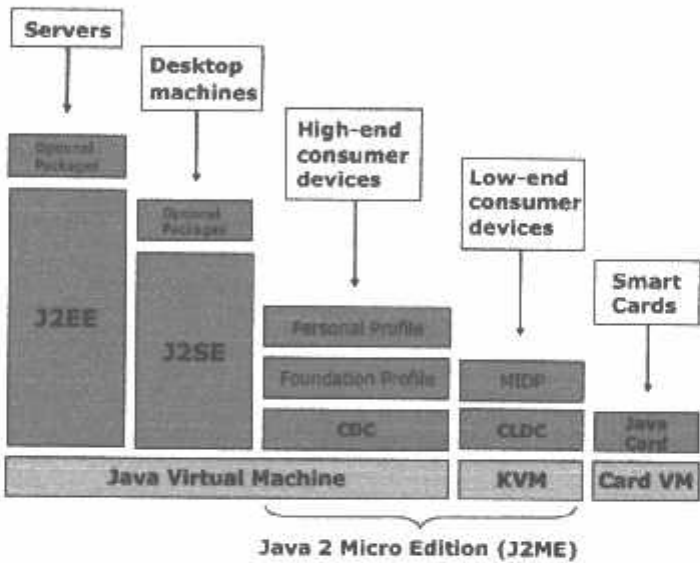
Java dibuat pada tahun 1991 oleh James Gosling. Pada awalnya diberi nama Oak, dimana untuk menghormati pohon yang ada di luar jendela Gosling. Kemudian namanya diubah ke Java karena telah ada sebuah bahasa yang diberi nama Oak. Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai platform yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan JAVA sebuah remote kontrol yang diberi nama Star 7.

Pada saat yang sama, World Wide Web dan Internet berkembang sangat cepat. Gosling menyadari bahwa Java dapat digunakan untuk pemrograman Internet. Dengan keluarnya versi 1.2, platform Java telah dipilah-pilah menjadi

beberapa edisi : The Standard Edition(J2SE), Enterprise Edition(J2EE), Mobile Edition(J2ME), dan JavaCard API.

Tabel 2.1. Platform Java

J2SE – Java 2 Platform, Standard Edition	Aplikasi Desktop
J2EE – Java 2 Platform, Enterprise Edition	Aplikasi enterprise dengan fokus pada pengembangan sisi webserver, termasuk servlet,JSP,EJB, dan XML
J2ME – Java 2 Platform, Micro Edition	Perangkat Mobile
JavaCard	Smart Cards



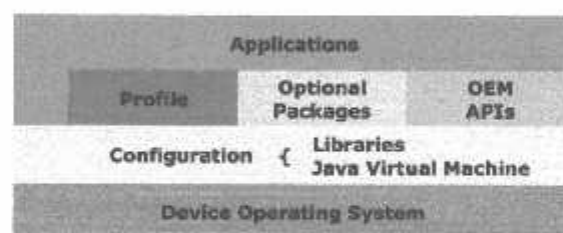
Gambar 2.1. Platform JAVA

2.2.1. J2ME

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah.

Dengan perkembangbiakan perangkat mobile konsumen dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada perangkat ini. Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program-program tersebut dikompile ke dalam bytecode dan diterjemahkan dengan Java Virtual Machine(JVM).Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

J2ME menyediakan suatu interface yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompile ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari J2ME terletak pada configuration dan profile-profile. Suatu configuration menggambarkan lingkungan runtime dasar dari suatu sistem J2ME. Ia menggambarkan core library, virtual machine, fitur keamanan dan jaringan.



Gambar 2.2. Arsitektur J2ME

Sebuah profile memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat. profile-profile menyediakan user interface(UI) API,persistence, messaging library, dan sebagainya. Satu set library tambahan atau package tambahan menyediakan kemampuan program tambahan. Pemasukan package ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak

memiliki Bluetooth built-in, sehingga Bluetooth API tidak disediakan dalam perangkat ini.

2.2.1.1. Configuration

Suatu configuration menggambarkan fitur minimal dari lingkungan lengkap Java runtime. Untuk menjamin kemampuan portabilitas dan interoperabilitas optimal diantara berbagai macam perangkat yang dibatasi sumber dayanya(memory, prosesor, koneksi yang dibatasi), configuration tidak menggambarkan fitur tambahan. Suatu configuration J2ME menggambarkan suatu komplemen yang minimum dari teknologi JAVA. Adalah merupakan tugas profile-profile untuk menggambarkan tambahan library untuk suatu kategori perangkat tertentu.

configuration menggambarkan:

- Subset bahasa pemrograman JAVA
- Kemampuan Java Virtual Machine(JVM)
- Core platform libraries
- Fitur sekuriti dan jaringan

2.2.1.2. Profile

Suatu profile menggambarkan set-set tambahan dari API dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara configuration menggambarkan library dasar, profile-profile menggambarkan library yang penting untuk membuat aplikasi-aplikasi efektif. Library ini memasukkan user interface, jaringan dan penyimpanan API.

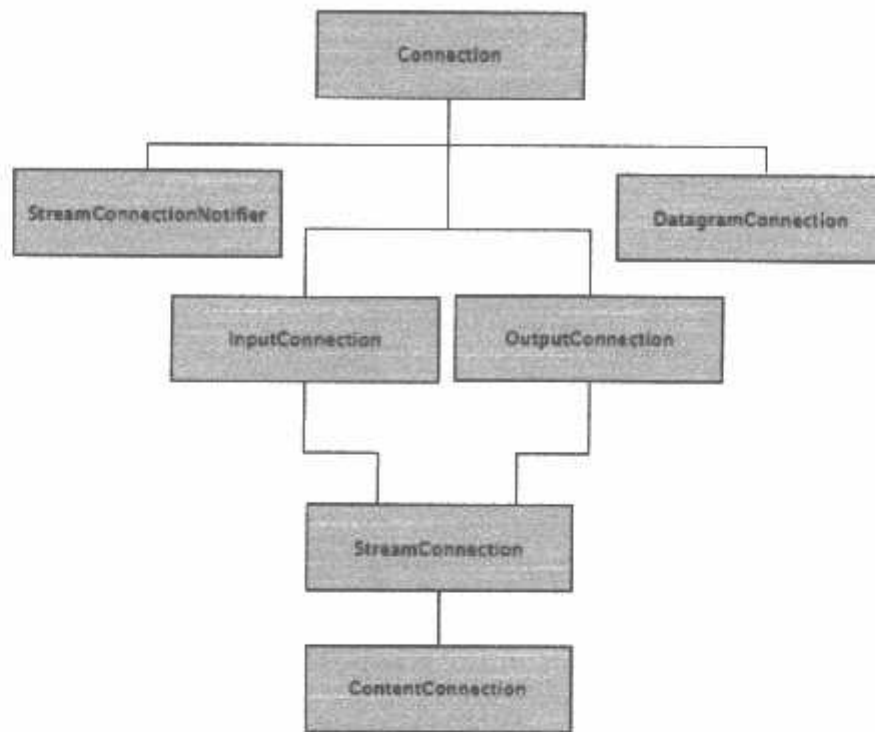
2.2.1.3. CLDC

The Connected Limited Device Configuration (CLDC) menggambarkan dan menunjuk pada area berikut ini:

- Fitur Bahasa Java dan Virtual Machine (VM)
- Library dasar (java.lang.*, java.util.*)
- Input/Output (java.io.*)
- Keamanan
- Jaringan
- Internationalization

2.2.1.4. Generic Connection Framework

The Generic Connection Framework menyediakan API dasar untuk koneksi dalam CLDC. Framework ini menyediakan suatu pondasi umum untuk koneksi seperti HTTP, Socket, dan Datagrams. GCF menyediakan suatu set API yang umum dan biasa yang memisahkan semua jenis koneksi. Perlu dicatat bahwa tidak semua jenis koneksi dibutuhkan untuk diterapkan oleh perangkat MIDP. Hirarki interface yang dapat diperluas dari GFC membuat proses penyamarataan menjadi mungkin. Jenis koneksi baru mungkin bisa ditambahkan ke dalam framework ini dengan memperluas hirarki ini.



Gambar 2.3. Hirarki Koneksi GCF (Generic Connection Framework)

2.2.1.5. MIDP

The Mobile Information Device Profile (MIDP) MIDP menggambarkan model aplikasi, UI API, penyimpanan dan jaringan yang kuat,

permainan dan media API, kebijakan keamanan, penyebaran aplikasi dan ketetapan over-the-air.

Spesifikasi MIDP menggambarkan suatu perangkat MID yang memiliki karakteristik

ini sebagai batas minimum:

❖ Tampilan:

Ukuran Layar: 96x54

kedalaman tampilan: 1-bit

Ketajaman pixel: sekitar 1:1

❖ Masukan:

Satu atau lebih mekanisme user-input: satu keyboard, dua keyboard, atau touch screen

❖ Memory:

256 kilobytes of non-volatile memory untuk implementasi MIDP.

8 kilobytes of non-volatile memory for application-created persistent data

128 kilobytes of volatile memory for the Java runtime (e.g., the Java heap)

❖ Jaringan:

dua jalur, wireless, bandwidth terbatas

❖ Sound:

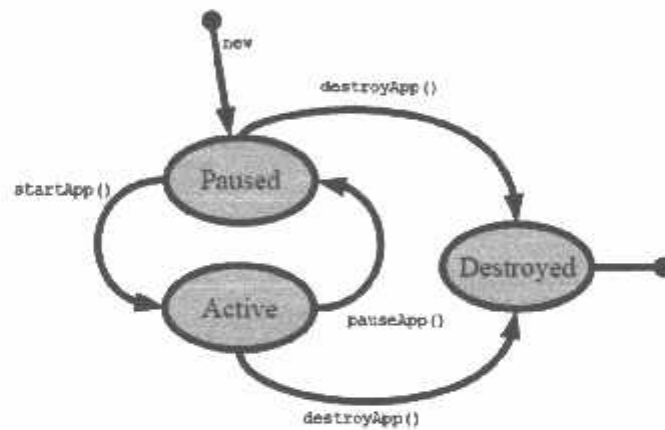
Kemampuan untuk memainkan nada-nada

2.2.1.6.Midlet

Suatu aplikasi MIDP disebut MIDlet. Perangkat application management software (AMS) berinteraksi langsung dengan MIDlet dengan method MIDlet create, start, pause, dan destroy. MIDlet adalah bagian dari package javax.microedition.midlet. Sebuah MIDlet harus di-extend dengan class MIDlet. Dan dapat meminta parameter dari AMS seperti dirumuskan dalam application descriptor (JAD). Suatu MIDlet tidak harus memiliki (dan memang harus tidak mempunyai) sebuah method public static void main(String[] argv).Method tersebut tidak akan dikenal lagi oleh AMS sebagai titik awal sebuah program.

2.2.1.7. 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.



Gambar 2.4. Siklus Hidup Midlet

2.3. Web Server Apache

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Apache memiliki fitur-fitur canggih

seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape (sekarang dikenal sebagai Sun Java Sistem Web Server). Sejak April 1996 Apache menjadi server web terpopuler di internet. Pada Mei 1999, Apache digunakan di 57% dari semua web server di dunia. Pada November 2005 persentase ini naik menjadi 71%. (sumber: Netcraft Web Server Survey, November 2005). Asal mula nama Apache berasal ketika sebuah server web populer yang dikembangkan pada awal 1995 yang bernama NCSA HTTPd 1.3 memiliki sejumlah perubahan besar terhadap kode sumbernya (patch). Karena banyaknya patch pada perangkat lunak tersebut sehingga disebut sebuah server yang memiliki banyak patch ("a patchy" server). Tetapi pada halaman FAQ situs web resminya, disebutkan bahwa "Apache" dipilih untuk menghormati suku asli Indian Amerika Apache (Indé), yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari Apache ditulis dari awal tanpa mengandung kode sumber dari NCSA.

2.4. PHP

adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing/Form Interpreter*. Dengan perilsan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

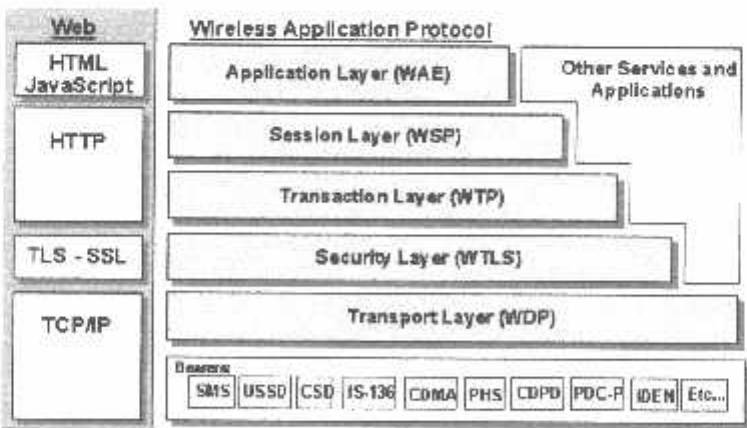
Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai

pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

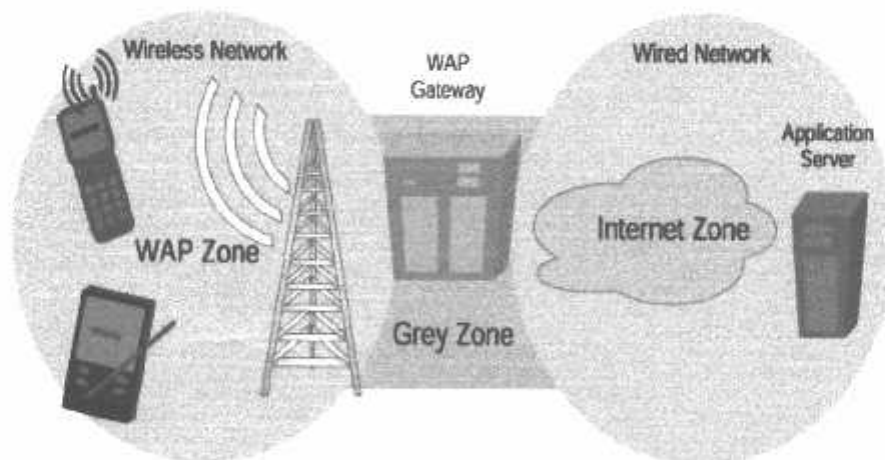
2.5. WAP

Wireless Application Protokol disingkat WAP adalah standar internasional terbuka untuk aplikasi yang menggunakan komunikasi nirkabel. Tujuan utamanya untuk membangun aplikasi yang dapat mengakses internet dari telepon genggam atau PDA. Ada beberapa versi WAP antara lain WAP 1.2.1 dan 2.0. WAP 1.2.1 hanya dapat menampilkan laman sederhana saja dibandingkan dengan WAP 2.0 yang mendukung bahasa xhtml dan gambar. WAP di buat pertama kali sebagai protokol komunikasi bergerak yang tidak bergantung pada sistem tertentu. WAP dirancang sebagai bagian dari sistem di masa depan sama halnya dengan Bluetooth dan GPRS. WAP merupakan protokol komunikasi bergerak yang terdiri dari beberapa layer dan dapat dijalankan pada sistem jaringan yang berbeda.



Gambar 2.5. Layer WAP

WAP sendiri terdiri dari 5 layer, yaitu Wireless Application Environment (WAE), Wireless Session Protocol (WSP), Wireless Transport Protocol (WTP), Wireless Data Protocol (WDP) dan Wireless Transport Layer Security (WTLS). Dari kelima layer tersebut yang terlihat hanyalah WAE, yang menampilkan isi web ke layar. WAE terdiri atas wireless markup language, sebuah variant HTML yang biasa digunakan untuk menampilkan isi web di monitor. WML terdiri dari text dan hyperlink, tetapi bukan grafis. Ketika halaman WAP terlihat di mini browser pada ponsel, bagian lain dari WAP menjalankan bagian peranan penting di belakang layar. Wireless Session Protocol membuka dan menutup koneksi ke situs WAP. Wireless Transport Protocol membuat paket data berjalan sesuai dengan arahnya. Koneksi wireless kurang dapat diandalkan dibandingkan dengan koneksi kabel. Sehingga untuk menjaga agar data yang anda kirim dan terima berjalan dengan benar sangatlah penting. Wireless Layer Security, sebuah bagian dari Secure Socket layer yang biasanya digunakan untuk keamanan di web, mengompres dan mengenkripsi data yang dikirim dari piranti wireless.



Gambar 2.6. Cara Kerja WAP

Saat kita terkoneksi ke jaringan wireless dan meminta akses ke sebuah situs web yang mendukung WAP, telepon seluler akan mengirimkan permintaan tersebut ke gelombang radio ke sel terdekat dan langsung dirutekan ke internet melalui sebuah server gateway. Server gateway tersebut menerjemahkan permintaan ke format standard HTTP dan meneruskannya ke situs web. Jika situs tersebut merespon, maka ia akan mengirimkan dokumen HTML ke server gateway kemudian dikoneksikan ke WML dan merutekannya ke antena terdekat. Antena tersebut akan mengirim via gelombang radio ke piranti WAP anda dan akhirnya micro browser akan menampilkan halaman tersebut.

2.6. GPRS

GPRS merupakan sistem transmisi berbasis paket untuk GSM yang menggunakan prinsip 'tunnelling'. Ia menawarkan laju data yang lebih tinggi. Laju datanya secara kasar sampai 160 kbps dibandingkan dengan 9,6kbps yang dapat disediakan oleh rangkaian tersakelar GSM. Kanal-kanal radio ganda dapat dialokasikan bagi seorang pengguna dan kanal yang sama dapat pula digunakan secara berbagi ('sharing') di antara beberapa pengguna sehingga menjadi sangat efisien. Dari segi biaya, pentarifan diharapkan hanya mengacu pada volume penggunaan. Penggunanya ditarik biaya dalam kaitannya dengan banyaknya byte yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan GPRS akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya daripada layanan-layanan IP. GPRS merupakan teknologi baru yang memungkinkan para operator jaringan komunikasi bergerak menawarkan

layanan data dengan laju bit yang lebih tinggi dengan tarif rendah ,sehingga membuat layanan data menjadi menarik bagi pasar massal. Para operator jaringan komunikasi bergerak di luar negeri kini melihat GPRS sebagai kunci untuk mengembangkan pasar komunikasi bergerak menjadi pesaing baru di lahan yang pernah menjadi milik jaringan kabel, yakni layanan internet. Kondisi ini dimungkinkan karena ledakan penggunaan internet melalui jaringan kabel (telepon) dapat pula dilakukan melalui jaringan bergerak. Sebagai gambaran kecil, layanan bergerak yang kini menjadi sukses di pasar (bagi operator di manca negara) misalnya adalah, laporan cuaca, pemesanan makanan, berita olah raga sampai ke informasi seperti berita-berita penting harian.

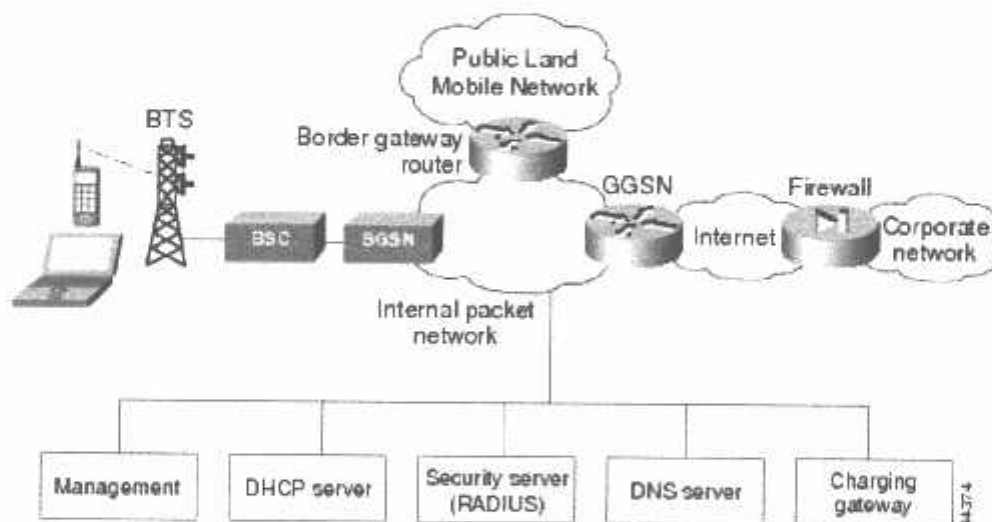
Dalam teorinya GPRS menjanjikan kecepatan mulai dari 56 kbps sampai 115 kbps, sehingga memungkinkan akses internet, pengiriman data multimedia ke komputer, *notebook* dan *handheld computer*. Namun, dalam implementasinya, hal tersebut sangat tergantung faktor-faktor sebagai berikut:

- Konfigurasi dan alokasi time slot pada level BTS
- Software yang dipergunakan
- Dukungan fitur dan aplikasi ponsel yang digunakan

Ini menjelaskan mengapa pada saat-saat tertentu dan di lokasi tertentu akses GPRS terasa lambat, bahkan lebih lambat dari akses CSD yang memiliki kecepatan 9,6 kbps.

Komponen-komponen utama jaringan GPRS adalah :

- GGSN (*Gateway GPRS Support Node*): gerbang penghubung jaringan GPRS ke jaringan internet. Fungsi dari komponen ini adalah sebagai interface ke PDN (Public Data Network), *information routing, network screening, user screening, address mapping*.
- SGSN (*Serving GPRS Support Node*): gerbang penghubung jaringan BSS/BTS ke jaringan GPRS. Komponen ini berfungsi untuk mengantarkan paket data ke MS, *update* pelanggan ke HLR, registrasi pelanggan baru.
- PCU : komponen di level BSS yang menghubungkan terminal ke jaringan GPRS



Gambar 2.7. Cara Kerja GPRS

SGSN bertugas : 1. Mengirim paket ke Mobile Station (MS) dalam satu area 2. Mengirim sejumlah pertanyaan ke HLR untuk memperoleh profile data pelanggan GPRS (*management mobility*) 3. Mendeteksi MS-GPRS yang baru dalam suatu

area servis yang menjadi tanggung jawabnya (location management) 4. SGSN dihubungkan ke BSS pada GSM dengan koneksi *Frame Relay* melalui PCU (Packet Control Unit) di dalam BSC

GGSN bertugas : 1. Sebagai *interface* ke jaringan IP external seperti : public internet atau mobile service provider 2. Meng-*update* informasi *routing* dari PDU (Protokol Data Units) ke SGSN.

GPRS menggunakan sistem komunikasi *packet switch* sebagai cara untuk mentransmisikan datanya. *Packet switch* adalah sebuah sistem di mana data yang akan ditransmisikan dibagi menjadi bagian-bagian kecil (paket) lalu ditransmisikan dan diubah kembali menjadi data semula. Sistem ini dapat mentransmisikan ribuan bahkan jutaan paket per detik. Transmisi dilakukan melalui PLMN (*Public Land Mobile Network*) dengan menggunakan IP *backbone*. Karena memungkinkan untuk pemakaian kanal transmisi secara bersamaan oleh pengguna lain maka biaya akses GPRS, secara teori, lebih murah daripada biaya akses CSD.

GPRS didesain untuk menyediakan layanan transfer packet data pada jaringan GSM dengan kecepatan yang lebih baik dari GSM. Kecepatan yang lebih baik ini didapat dengan menggunakan *coding scheme* (CS) yang berbeda dari GSM.

BAB III

DESAIN DAN ANALISA SISTEM MOBILE BANKING

Dalam bab ini akan dibahas desain dan analisa sistem. Analisa akan dilakukan pada setiap blok desain, Desain disini yang dibuat adalah desain dari mobile application dalam melakukan kegiatan perbankan.

Server meliputi database, dan servlet penerima request dari client, sedangkan client meliputi content Mobile Banking, analisa masing-masing blok desain, secara garis besar terdapat dua bagian analisa yaitu : Analisa secara Hardware, Analisa secara Software.

Pada analisa secara hardware meliputi cara kerja komunikasi dari handphone hingga ke server secara hardware pada setiap blok desain, sedangkan analisa secara software meliputi algoritma & tahapan program yang digunakan pada setiap blok desain.

3.1. Spesifikasi Sistem

Sistem Mobile banking yang dibangun melakukan komunikasi antara hardware (Handphone) dimana di dalam handphone di install aplikasi mobile banking yang di desain, dari aplikasi itulah user melakukan komunikasi dengan web server, melakukan berbagai kegiatan transaksi sesuai dengan konten yang ada, di serverpun ada aplikasi untuk menerima paket data dari user mengolah, membandingkan dengan database lalu mengirimkan kembali pada user. Maka adapun spesifikasi hardware dan software pada system Mobile banking.

3.1.1. Spesifikasi Software

Dalam melakukan desain mobile banking di butuhkan beberapa software pendukung yang mampu melengkapi dan mendukung kinerja mobile banking yang di desain baik itu di sisi server ataupun di sisi klien, adapun spesifikasi software di sisi server dan client sebagai berikut: Server side (Web Server), dan Client side (Development application & Testing application).

➤ Server Side

Web Server Apache

Untuk mewujudkan koneksi online maka di beli sebuah domain dan hosting di salah satu hosting Indonesia yaitu rumahWeb,dimana di dalam web server ini nantinya di simpan data-data user dan servlet untuk komunikasi dengan user, server ini bersistem operasi unix adapun Spesifikasi Hosting yang dimiliki adalah sebagai berikut :

- Web Server : Apache 2.2.9
- Mysql Version : 5
- PHP Version : 5.5.2
- Support : CGI, SSI, Perl
- Mod_SSL : 2.2.9

Web Server inilah yang nantinya di akses oleh aplikasi mobile banking, server ini di program untuk bisa menerima, memproses dan membandingkan dengan database dan perintah yang ada sesuai dengan permintaan yang di kirim oleh user, dan nantinya hasil dari proses dikirimkan lagi ke user mobile banking.

➤ Client Side

Development Application

Dalam mendesain mobile banking di butuhkan software pembangun aplikasi yang terdiri dari :

- JDK (Java Development Kit)

Lingkungan pembangun java yang di gunakan adalah J2SDK 1.6 dimana selain terdapat tool untuk develop aplikasi java juga terdapat Java Runtime Environment yaitu sebuah lingkungan untuk menjalankan aplikasi java

- JTWK (Java Wireless Tool Kit)

Sebuah lingkungan khusus dari java untuk membangun aplikasi wireless dalam hal ini aplikasi yang berjalan pada device yang memiliki portabilitas tinggi adapun JTWK yang digunakan adalah JTWK dari vendor Motorola, karena nantinya aplikasi akan dilakukan implementasi pada mobile device merk Motorola, oleh karena itu kita butuhkan JTWK dari Motorola agar aplikasi yang kita buat sesuai dengan spesifikasi device Motorola, JTWK Motorola yang di pakai adalah Motorola J2ME(TM) SDK v6.1.1 for Motorola OS Products.

Testing Application

Setelah dilakukan pembangunan software maka dibutuhkan emulator untuk testing aplikasi sebelum kita testing pada hardware yang sesungguhnya ini dilakukan agar jika terjadi kesalahan error program tidak sampai mengganggu kinerja hardware, dan kesalahan bisa di tangani lebih awal, emulator yang digunakan sesuai dengan mobile device yang nantinya akan di gunakan yaitu

Motorola e-398 yang sudah include pada Motorola J2ME(TM) SDK v6.1.1 for Motorola OS Products.



Gambar 3.1. Emulator e-398

3.1.2. Spesifikasi Hardware

Setelah aplikasi yang dibuat telah selesai maka nantinya dilakukan implementasi pada hardware yang sesungguhnya, hardware yang di pakai adalah Handphone Motorola e-398 dimana Handphone tersebut kita install aplikasi berekstensi **.jar**, adapun spesifikasi untuk lingkungan JAVA pada Motorola e-398 sebagai berikut :

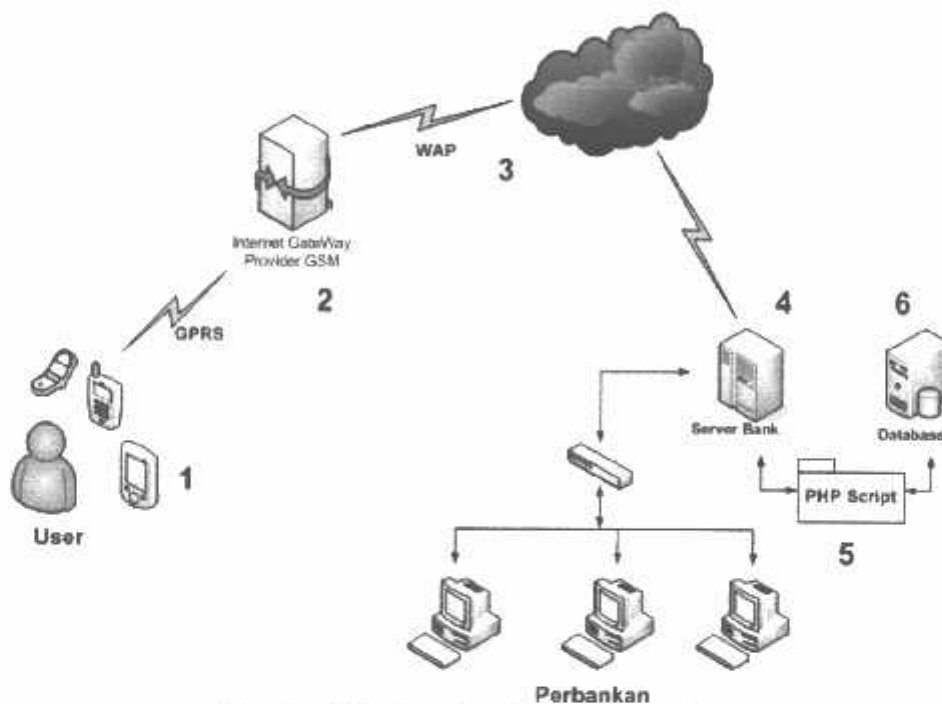
- CLDC Version : 1.1
- MIDP Version : 2.0



Gambar 3.2. Motorola e-398

3.2. Desain Keseluruhan Sistem

Perancangan desain Mobile Banking agar dapat dilakukan secara sistematis dan terstruktur maka perlu dibuat blok sistem yang menjelaskan dari sistem yang dirancang. Secara garis besar perancangan desain sistem ditunjukkan pada gambar berikut :



Gambar 3.3. Desain Keseluruhan Sistem

Keterangan :

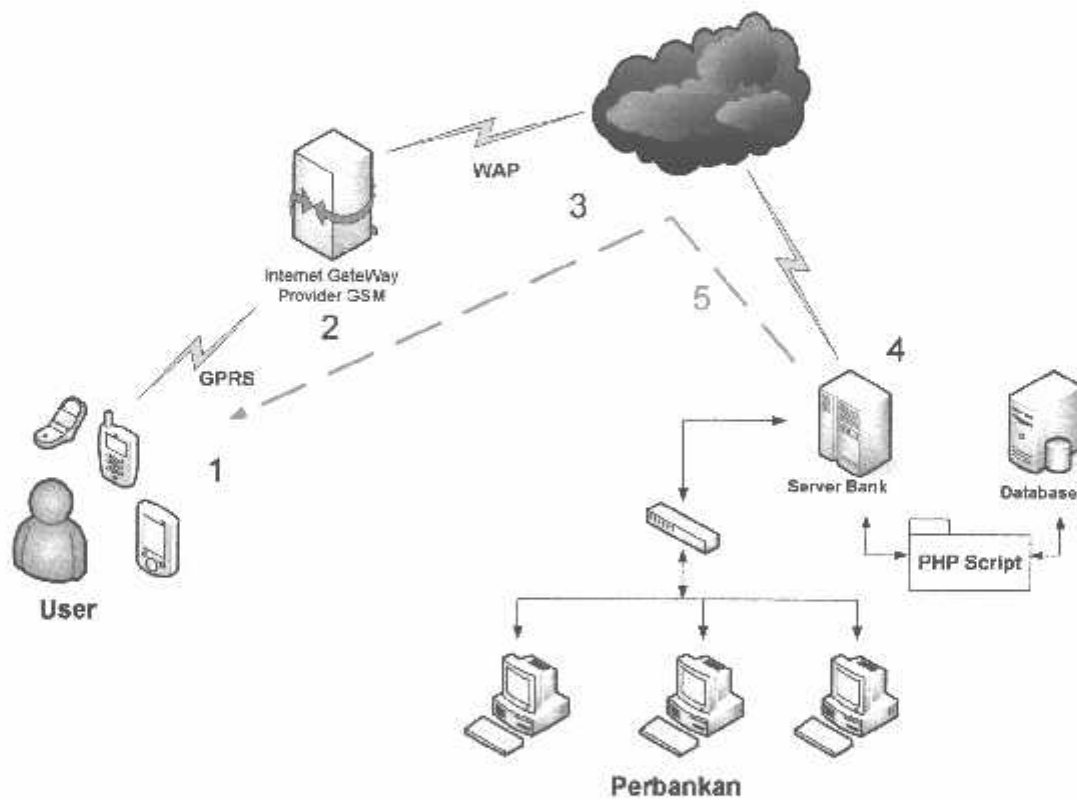
1. Aplikasi User yang di install pada handphone user.
2. Provider GSM yang menterjemahkan sinyal GPRS menjadi request HTTP.
3. Protokol yang digunakan untuk berkomunikasi dengan Internet.
4. Server bersistem oprasi UNIX, dengan Web Server Apache.
5. PHP sebagai pengolah dan penghubung database.
6. Database Mysql untuk menyimpan data user.

3.2.1. Desain Komunikasi HTTP

Web Server membuka port 80 untuk berkomunikasi dengan dunia luar, maka dalam melakukan komunikasi dengan web server aplikasi user memanfaatkan port tersebut, dimana request dilakukan dengan memetakan URL yang telah ditanam pada script JAVA dan melakukan request dengan metode GET.

➤ Analisa Hardware

Analisa Hardware ini membahas aliran koneksi secara hardware dimana aliran data melewati beberapa proses sebelum diterima oleh server, adapun aliran sebagai berikut;



Gambar 3.4. Analisa Komunikasi HTTP secara Hardware

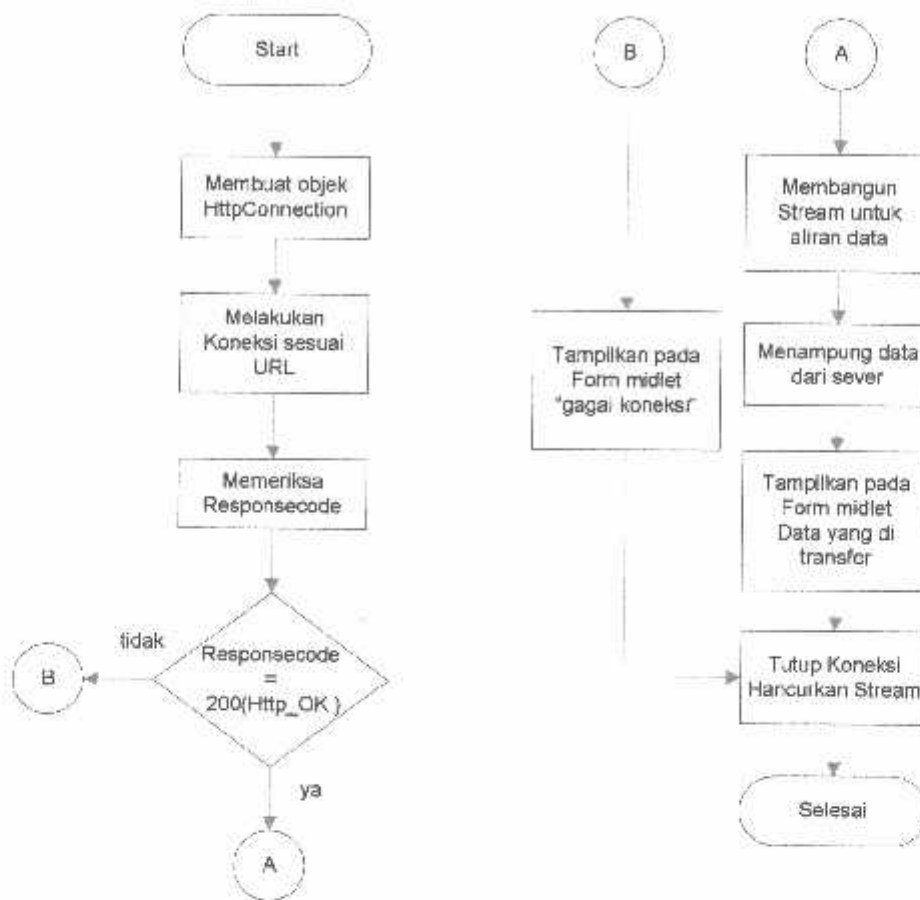
Keterangan dari Desain sistem di atas:

1. User dengan handphone support J2ME di install aplikasi mobile banking, lalu melakukan request ke server bank, dengan mengakses port HTTP 80 dalam komunikasi dengan server bank teknologi transfer data yang digunakan pada provider GSM adalah GPRS, awalnya request dari user adalah berupa sinyal GPRS.
2. Pada Provider GSM request dari user yang berupa sinyal GPRS akan di terjemahkan menjadi request berformat HTTP yang nantinya berkomunikasi dengan internet melalui protocol wireless (WAP).
3. Request dari user yang telah di terjemahkan oleh provider ke dalam format HTTP lalu di alamatkan sesuai dengan URL yang telah dikodekan dalam aplikasi user, dalam hal ini user melakukan request ke sebuah web server yang telah di persiapan di salah satu hosting Indonesia.
4. Server HTTP membuka port 80 untuk berkomunikasi dengan pihak lain, dalam kondisi normal maka server bersikap listening, menunggu paket yang masuk, setelah paket dari user diterima oleh server maka server akan memberikan respon berupa koneksi oke (HTTP_OK 200), setelah aplikasi menerima respon tersebut maka aplikasi user pada java membentuk hubungan khusus dalam hal ini Stream, untuk aliran data khusus antara user dan server.
5. Lalu server mereturn replay ke user melalui route internet untuk dikembalikan pada alamat pengirim dalam hal ini provider GSM, masih lewat hubungan Stream yang di ciptakan java, lalu dari GSM selanjutnya

di terjemahkan kembali dalam bentuk sinyal GPRS dan di terima kembali oleh user, setelah user menerima return dari server maka hubungan stream tadi di hancurkan oleh DestroyApp pada packet JAVA.

➤ Analisa Software

Komunikasi dengan web server yang dilakukan oleh aplikasi user dalam hal ini lebih mengacu pada pembangunan software J2ME, dalam melakukan komunikasi dengan server algoritma yang digunakan adalah dengan melemparkan paket ke sebuah server sesuai dengan URL. aplikasi setelah itu jika server itu ada maka server tersebut akan memberikan nilai kembalian berupa responsecode jika response code = 200 itu berarti koneksi bisa dilakukan, tetapi jika bukan 200 maka koneksi tidak bisa dilakukan, dalam melakukan koneksi tersebut program aplikasi memanfaatkan API dari java yang support dalam melakukan HttpURLConnection, dimana koneksi akan diciptakan dari melalui objek untuk mengakses API komunikasi, API inilah yang nantinya menangani komunikasi Http terlihat pada flowchart berikut:



Gambar 3.5. Flowchart Komunikasi Http

Tahapan yang dilakukan dalam melakukan komunikasi dengan web server Http sebagai berikut:

0. Inisialisasi Sistem.
1. Membangun objek dari kelas HttpURLConnection, dimana kelas ini berfungsi untuk mengatur koneksi Http.
2. Membangun koneksi ke suatu alamat sesuai dengan URL (Uniform Locator Resource).
3. Dalam melakukan komunikasi dengan server, maka sebelum melakukan komunikasi sesungguhnya ada yang dinamakan pengecekan untuk mengecek apakah alamat yang di tuju sesuai URL benar-benar ada selain itu juga untuk mengecek kondisi jaringan, biasanya server memberikan

respon berupa code untuk menggambarkan kondisi yang ada di server
adapun beberapa respon code dari server sebagai berikut :

Tabel 3.1. Response code Koneksi HTTP

Code	Response
200	<u>HTTP OK</u>
203	<u>HTTP NOT AUTHORITATIVE</u>
204	<u>HTTP NO CONTENT</u>
305	<u>HTTP USE PROXY</u>
400	<u>HTTP BAD REQUEST</u>
403	<u>HTTP FORBIDDEN</u>
404	<u>HTTP NOT FOUND</u>
405	<u>HTTP RAD METHOD</u>
406	<u>HTTP NOT ACCEPTABLE</u>
408	<u>HTTP CLIENT TIMEOUT</u>
409	<u>HTTP CONFLICT</u>
410	<u>HTTP GONE</u>
413	<u>HTTP ENTITY TOO LARGE</u>
414	<u>HTTP REQ TOO LONG</u>
415	<u>HTTP UNSUPPORTED TYPE</u>
416	<u>HTTP UNSUPPORTED RANGE</u>
417	<u>HTTP EXPECT FAILED</u>
500	<u>HTTP INTERNAL ERROR</u>
501	<u>HTTP NOT IMPLEMENTED</u>
502	<u>HTTP BAD GATEWAY</u>
503	<u>HTTP UNAVAILABLE</u>
504	<u>HTTP GATEWAY TIMEOUT</u>

4. Jika Response code yang di kirim oleh server buka 200 maka bisa dikatakan koneksi yang terjadi tidak bisa di lanjutkan karena bermasalah, kita tidak perlu mengetahui apa masalahnya yang jelas jika response code bukan 200 maka koneksi putus dan java membangkitkan error message pada form berupa "Gagal Koneksi", sebaliknya jika response code yang di kirim oleh server adalah 200 maka koneksi bisa dilakukan dan aplikasi java user bisa berkomunikasi dengan web server.

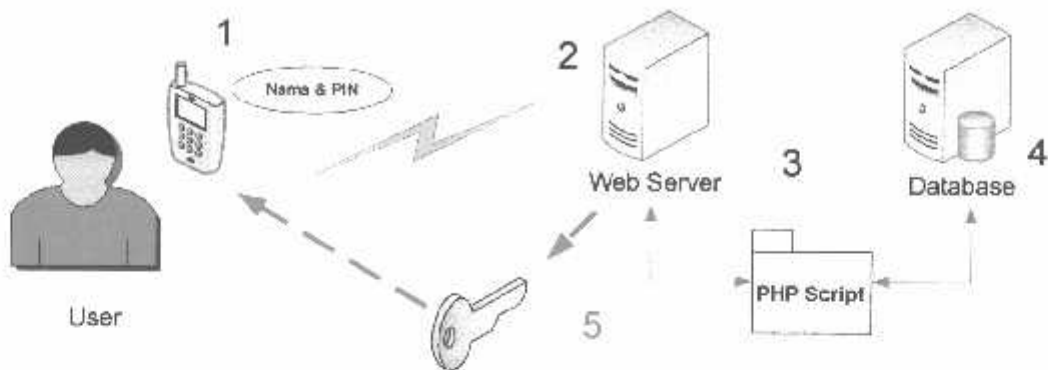
5. Setelah server mengirim response code dan code itu bernilai 200 maka java membuat suatu stream dari kelas **InputStreamReader** yang berfungsi untuk melakukan aliran data khusus koneksi ini saja.
6. Setelah Stream terbentuk maka server menerima request dari user dan memprosesnya, setelah server memproses maka server mengirimkan hasil dari proses, yang mana kiriman dari server tersebut ditampung dalam Buffer yang terdapat dalam **InputStreamReader**.
7. Pemeriksaan terhadap kiriman server, apakah kiriman server berisi data atau tidak.
8. Jika hasil dari server kosong atau tidak berisi apa-apa bisa saja akses terhadap file php tujuan user tidak ada dan pada saat itu akan di hangkitkan error message oleh java, sedangkan jika kiriman dari server berisi data maka ditampilkan pada form midlet.
9. Setelah request dari user di terima dan proses pengiriman hasil proses server telah di dapatkan oleh aplikasi user maka hubungan untuk satu koneksi ini di tutup dan stream otomatis di hancurkan.

3.2.2. Desain Transaksi

Dalam melakukan transaksi antara user dan sever di bentuklah suatu system pertukaran kunci untuk menjamin keamanan, dimana sebuah fungsi transaksi hanya bisa dibuka jika dikirimkan key dari server, adapun system sebagai berikut:

➤ Analisa Hardware

Analisa secara hardware ini menjelaskan bagaimana aliran sebuah transaksi secara hardware, adapun desain sebagai berikut:



Gambar 3.6. Pertukaran Key oleh Server

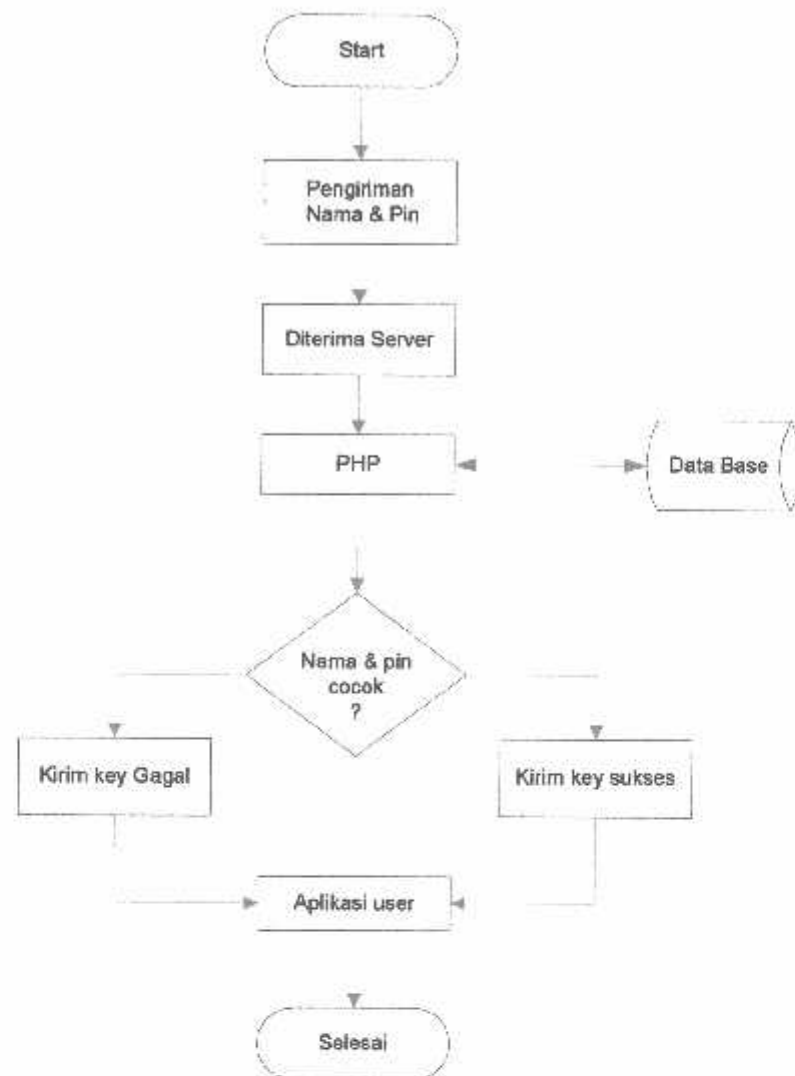
Keterangan :

1. Dalam melakukan transaksi user mengirimkan nama & pin ke server yang sebelumnya dilakukan enkripsi.
2. Server Mencrima nama & pin tersebut lalu di lanjutkan ke PHP.
3. PHP Script yang berfungsi sebagai servlet melakukan pemrosesan dengan database.
4. nama & pin yang di kirim oleh PHP di bandingkan ke database, lalu PHP memproses hasil.
5. Setelah dip roses oleh PHP maka server mengirimkan kunci/key sesuai dengan hasil proses, jika data nama & pin tidak sesuai dengan database maka kunci / key yang dikirim adalah kunci berisi pesan gagal, sedangkan jika nama & pin sesuai maka kunci yang dikirim berisi pesan sukses.

setelah diterima oleh user maka , jika pesan berisi gagal maka user tidak bisa mengakses bagian transaksi, sedangkan jika kunci / key yang dikirim berisi pesan sukses maka user bisa mengakses bagian transaksi pada aplikasi midlet user.

➤ **Analisa Software**

Algoritma yang digunakan untuk melakukan transaksi dengan webserver memanfaatkan kelas komunikasi input-output dalam paket – paket yang telah disediakan oleh platform java, meskipun demikian kita tetap harus melakukan sorting terhadap kelas – kelas tersebut agar hanya kelas – kelas yang dibutuhkan saja yang di eksekusi, jika tidak maka akan berakibat selain error juga mengurangi kinerja dari aplikasi itu sendiri, oleh karena itu hanya beberapa kelas saja yang mendukung koneksi ini yang di bangun objeknya, adapun flowchart dari proses transaksi sebagai berikut.



Gambar 3.7. Flowchart transaksi

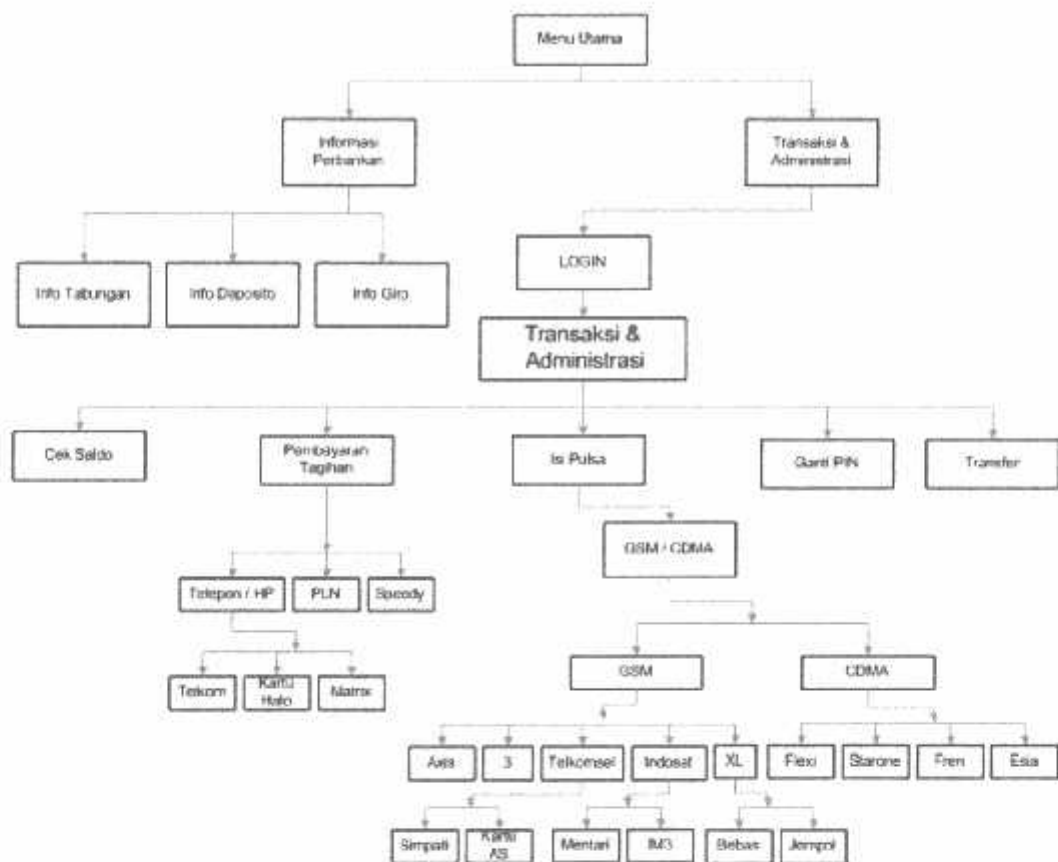
Tahapan yang dilakukan dalam melakukan transaksi dengan web server Http sebagai berikut:

0. Inisialisasi sistem.
1. Pengiriman paket nama & pin oleh aplikasi user ke server.
2. Server menerima paket dari user lalu diteruskan ke PHP.
3. PHP melakukan pemrosesan sesuai dengan paket yang masuk lalu melakukan pengecekan dengan database.

4. Pemeriksaan apakah nama & pin sesuai dengan database yang ada ?
5. Jika sesuai kirimkan kunci berisi sukses, jika tidak sesuai kirimkan kunci berisi gagal.
6. Aplikasi menerima kunci dari server dan melakukan kegiatan sesuai kunci tersebut, jika kunci berisi pesan gagal maka user tidak bisa melakukan kegiatan transaksi, sedangkan jika kunci berisi pesan sukses maka user bisa melakukan kegiatan transaksi.

3.2.3. Desain Konten

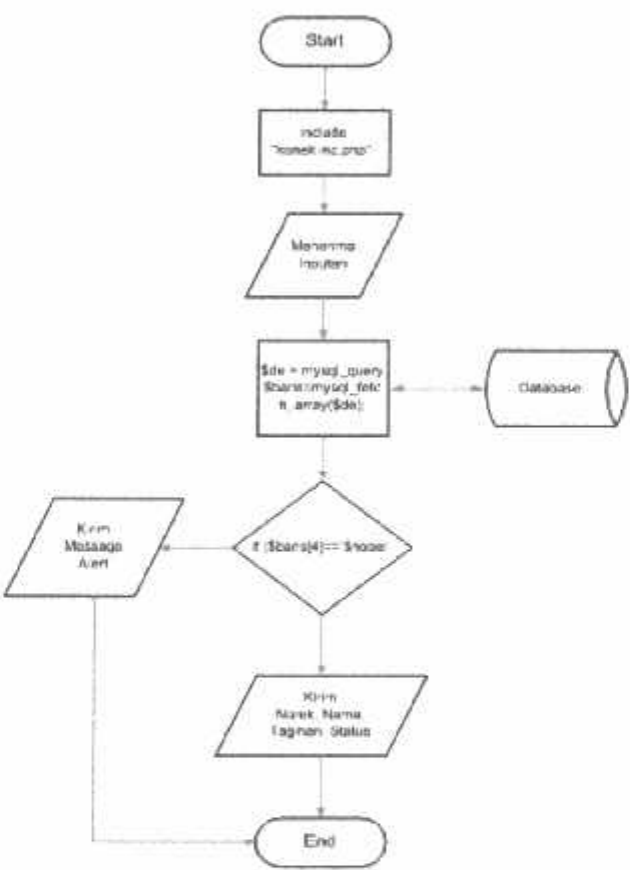
Dalam sebuah aplikasi Mobile ada hal yang terpenting yang harus di perhatikan yaitu konten, jika kita berbicara konten tentunya konten ini harus sesuai dengan kebutuhan dan keinginan pasar jangan sampai ada konten yang sia-sia saja, oleh karena itu di rancanglah konten mobile banking yang dapat memenuhi segala kegiatan transaksi perbankan, dimana dengan konten-konten inilah kegiatan perbankan lebih terorganisir, karena setiap kebutuhan telah di desain sedemikian rupa sehingga memudahkan user dalam melakukan kegiatan transaksi perbankan secara mobile, adapun rancangan konten dalam aplikasi mobile banking sebagai berikut :



Gambar 3.8. Desain Konten Mobile Banking

3.2.4. Desain Server

Dalam Melakukan komunikasi dengan server, aplikasi mobile banking melakukan request pada server dan server memberikan reply pada aplikasi mobile banking, request yang dilakukan oleh aplikasi mobile banking diolah terlebih dahulu oleh script PHP dan dilakukan koneksi ke database, adapun flowchart php di sisi server sebagai berikut :



Gambar 3.9. Desain servlet server

3.2.5. Desain Database

Dalam mengolah *request* dari aplikasi mobile banking, server melakukan komunikasi dengan database untuk mengambil data dan menyimpan data, dari desain konten yang ada maka dibutuhkan beberapa database untuk memenuhi kebutuhan dari desain konten, seperti pembayaran tagihan telepon, PLN dll, berikut desain database yang berada di web server.

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	20	0	☐	
Pin	int	5	0	☐	
Saldo	int	10	0	☐	
NoRekening	int	10	0	☐	
Status	int	1	0	☐	

Gambar 3.10. Tbl_data_user

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	15	0	☐	
Tagihan	int	10	0	☐	
Status	varchar	11	0	☐	
Pin	int	10	0	☐	
NoPelanggan	varchar	10	0	☐	

Gambar 3.11. Tbl_halo

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	10	0	☐	
Tagihan	int	10	0	☐	
Status	varchar	11	0	☐	
Pin	int	10	0	☐	
NoPelanggan	varchar	10	0	☐	

Gambar 3.12. tbl_matrix

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	15	0	☐	
Tagihan	int	10	0	☐	
Status	varchar	11	0	☐	
Pin	int	10	0	☐	
NoPelanggan	int	10	0	☐	

Gambar 3.13. tbl_pln

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	15	0	☐	
Tagihan	int	10	0	☐	
Status	varchar	11	0	☐	
Pin	int	10	0	☐	
NoPelanggan	int	10	0	☐	

Gambar 3.14. tbl_speedy

Name	Type	Length	Decimals	Allow Null	
► Nama	varchar	14	0	☐	
Tagihan	int	10	0	☐	
Status	varchar	11	0	☐	
Pin	int	10	0	☐	
NoPelanggan	int	10	0	☐	

Gambar 3.15. tbl_telkom

BAB IV

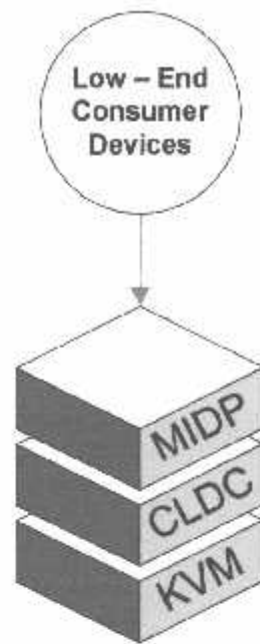
IMPLEMENTASI DAN PENGUJIAN SISTEM MOBILE BANKING

Sebagai cara untuk mengetahui unjuk kerja dari sistem yang dirancang dan dibuat, maka dilakukan implementasi dan pengujian sistem. Implementasi meliputi install aplikasi mobile banking pada handphone Motorola e-398 dan juga pengisian server dengan PHP servlet dan database. Pengujian meliputi koneksi antara aplikasi di handphone dengan aplikasi server, selain itu juga pengujian konten yang ada.

4.1. Pengujian Koneksi HTTP

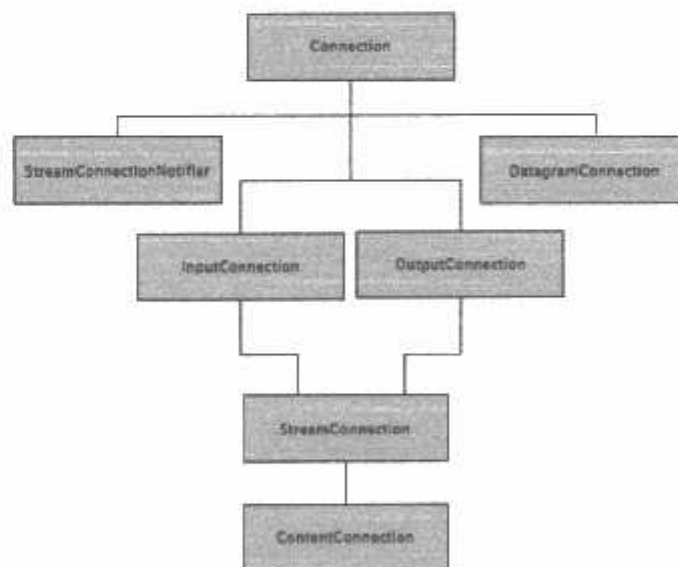
Pengujian Koneksi HTTP dilakukan dengan menggunakan Handphone Motorola e-398 yang telah di install Aplikasi Mobile Banking, yang mana nantinya melakukan request ke web server, dengan metode HEAD untuk mendapatkan informasi header dari sebuah web server, jika kita mendapatkan response maka web server akan mengirimkan tentang informasi dari web server, dengan begitu bisa dikatakan bahwa komunikasi antara aplikasi mobile banking dengan web server bisa dilakukan.

Dalam Melakukan request kepada web server, java dalam hal ini J2ME (Low end – Consumer User) menggunakan MIDP versi 2.0 yang mengacu pada display layar yang telah di dukung oleh Motorola e-398, CLDC untuk komunikasi jaringan mengacu pada CLDC, dimana dalam CLDC menggambarkan salah satunya jaringan.



Gambar 4.1. Arsitektur Low – End Consumer Devices

API dasar untuk koneksi CLDC disediakan oleh sebuah Framework dimana didalamnya telah tersedia berbagai koneksi datagram, Stream dll framework itu adalah GFC (Generic Connection Framework)

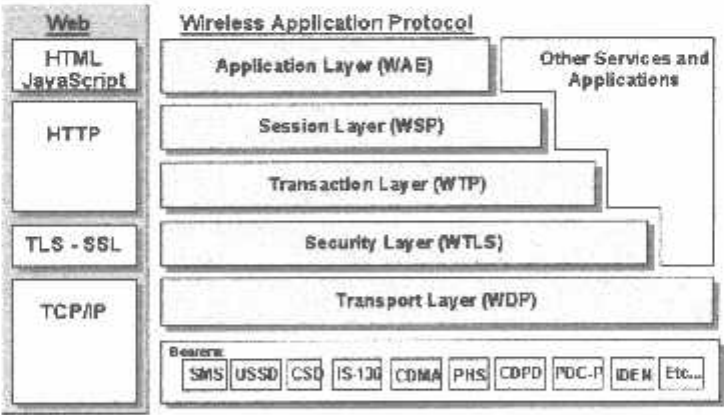


Gambar 4.2. Hirarki GCF

Dalam membuka koneksi kita bisa menggunakan `Connector.open()` dan mengcastingnya dengan salah satu dari ketiga interface berikut : `StreamConnection`, `ContentConnection`, `HttpConnection`, jika pada `StreamConnection` kita tidak bisa menentukan panjang dari sebuah replay, berbeda dengan `HttpConnection` dan `ContentConnection` dimana panjang replay bisa kita tentukan, interface yang digunakan dalam koneksi HTTP berikut adalah `HttpConnection`.

`HttpConnection` memiliki banyak sekali fungsi yang bias kita manfaatkan, setelah kita rujuk interface `HttpConnection`, maka paket kita lempar ke web server, dalam hirarki perjalannya paket dilempar berupa sinyal GPRS lalu di terjemahkan oleh provider menjadi standart request HTTP dan di teruskan ke web server menggunakan WAP.

Transfer paket yang dilakukan oleh provider ke jaringan global internet melalui WAP (Wireless Application Protokol), dimana paket yang di alirkan melalui Session Protokol yang di gunakan untuk membuka koneksi jaringan, Aliran paket dari jaringan di terima dan dikirim melalui datagram protocol.



Gambar 4.3. Layer WAP

adapun source code sebagai berikut :

```
private String[][] getInfo(String url) throws IOException {
    //1. Buat objek koneksi HTTP dari interface HttpConnection
    HttpConnection con = null;
    int responsecode;

    try {
        con = (HttpConnection) Connector.open(url);
        //2. Atur metode Request dan properti-propertinya dalam hal ini
        request HEAD
        con.setRequestMethod(HttpConnection.HEAD);
        con.setRequestProperty("User-Agent", "Profile/MIDP-2.0
        Configuration/CLDC-1.0");
        con.setRequestProperty("Content-Language", "en-us");

        //3. Periksa kode response dari web server
        // Bila kode tidak sama dengan 200 (HTTP_OK), bangkitkan
        error.
        responsecode = con.getResponseCode();
        if (responsecode != HttpConnection.HTTP_OK) {
            throw new IOException("HTTP Error with code: " +
            responsecode);
        }
        //4. cari jumlah field HTTP header
        int nField = 0;
        while (con.getHeaderField(nField) != null) {
            nField++;
        }
        //5. Pindahkan field HTTP header ke array string 2 dimensi
        String[][] field = new String[2][nField];
        for (int i=0; i<nField; i++) {
            field[0][i] = con.getHeaderFieldKey(i);
            field[1][i] = con.getHeaderField(i);
        }
        return field;
    } catch (Exception e) {
        throw new IllegalArgumentException(e.getMessage());
    } finally {
        //6. Tutup koneksi HTTP
        if (con != null) {
            con.close();
        }
    }
}
```

```

class infoHeaderWebServer extends Thread {
    private CommandListener listener;
    public void setListener (CommandListener cl) {
        listener = cl;
    }
    public void run() {
        try {
            // Panggil method getInfo dengan memparsing parameter
            URL
            // Data URL didapatkan dari txtURL
            String[][] aField = getInfo(txtURL.getString());
            for(int i=0;i<aField[1].length;i++) {
                frmOutput.append(aField[0][i] + "\n " + aField[1][i] +
                "\n");
            }
        } catch(Exception e) {
            frmOutput.append("Gagal ambil informasi header web
            server");
        }

        frmOutput.addCommand(get_cmdExit());
        frmOutput.setCommandListener(listener);
        getDisplay().setCurrent(frmOutput),
    }
}

```

Hasil dari source code diatas adalah sebuah informasi header web server yang memberikan informasi tentang web server yang kita rujuk, dalam hal ini adalah web server mobile banking.



Gambar 4.4. Hasil Komunikasi Http

4.2. Pengujian Login

Pengujian Login dilakukan untuk merequest key pada server, nama dan pin sebelum dikirim dilakukan enkripsi terlebih dahulu, jika nama dan pin yang dimasukkan sesuai dengan database, maka akan di kirim key/kunci untuk membuka aplikasi transaksi, jika nama dan pin tidak sesuai dengan database maka dikirim key/kunci untuk menyatakan kesalahan login.

Source code :

```
class login extends Thread {
    private CommandListener listener;
    public void setListener (CommandListener cl) {
        listener = cl;
    }
    public void run() {
        String rs="";
        user = txt_Nama.getString();
        String pwd = txt_Passwd.getString();

        try {
            // Panggil method signPHP
            rs = sign(user,pwd);
        }
        catch(Exception err) {
            System.out.println(err);
            Frm_Gagal.append("gagal koneksi!");
        }
        finally {
            // Proses user yang melakukan login

            if(rs.equals(key)){
                getDisplay().setCurrent(get_Frm_Sukses());
            }
            else {
                getDisplay().setCurrent(get_Frm_Gagal());
            }
        }
    }
}
```

Pada saat login akan diperiksa apakah textField nama & pin sudah terisi atau belum, untuk menghindari kesalahan login, pada textField pin, di atur sehingga hanya numeric saja yang bisa di inputkan, dan di ubah karakternya menjadi bintang.



Gambar 4.5. Login & Periksa Data



Gambar 4.6. Login Sukses & Gagal

4.3. Pengujian Isi Pulsa

Dalam Melakukan Pengisian Pulsa user di hadapkan pada pemilihan jaringan GSM / CDMA untuk lebih mempermudah pengisian pulsa, pengisian nomor handphone pada textField diperiksa terlebih dahulu oleh aplikasi apakah sudah terisi atau belum untuk menghindari salah pengiriman, jika ternyata kosong maka aplikasi akan memberi warning, setelah nomor Handphone tujuan dan nominal pulsa di masukkan maka di lakukan konfirmasi terlebih dahulu sebelum data di kirim ke server, setelah setuju dengan konfirmasi maka permintaan user di kirim

ke server, di server permintaan user di olah dahulu untuk di bandingkan dengan jumlah saldo yang ada, jika saldo mencukupi permintaan di proses, sedangkan jika saldo tidak mencukupi maka permintaan akan di tolak.



Gambar 4.7. Pengisian pulsa & Periksa Data



Gambar 4.8. Konfirmasi Pengisian Pulsa

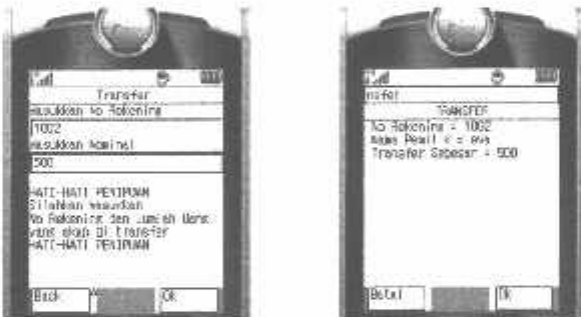


Gambar 4.9. Saldo Mencukupi & Saldo Tidak Mencukupi (isi Pulsa)

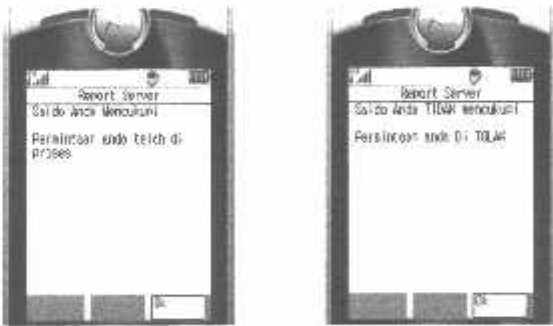
4.4. Pengujian Transfer

Pengujian Transfer dilakukan untuk meneguji apakah aplikasi sudah berjalan dengan benar, dalam melakukan transfer di butuhkan berupa inputan nomor

rekening dan jumlah nominal dari user, oleh karena itu inputan dari user sebelum dikirim ke server dilakukan pengecekan apakah sudah memasukkan nomor rekening dan nominal untuk menghindari kekosongan inputan, jika nomor rekening dan jumlah nominal belum dimasukkan maka aplikasi tidak akan mengirim data tersebut ke server, setelah user mengisi no rekening tujuan dan jumlah nominal, maka dilakukan maka di lakukan konfirmasi terlebih dahulu pada server untuk mengetahui lebih jelas siapa pemilik nomor rekening tujuan kita, untuk menghindari salah kirim, setelah mendapat data pemilik nomor rekening yang akan kita transfer, maka selanjutnya maka permintaan user atas transfer di kirim ke server, di server permintaan user di olah dahulu untuk di bandingkan dengan jumlah saldo yang ada, jika saldo mencukupi permintaan di proses, sedangkan jika saldo tidak mencukupi maka permintaan akan di tolak.



Gambar 4.10. Transfer & Konfirmasi



Gambar 4.11. Saldo Mencukupi & Saldo Tidak Mencukupi (Transfer)

4.5. Pengujian Cek Saldo

Pengecekan saldo dilakukan dengan menekan tombol refresh maka data akan di update langsung dari server, dengan begitu user bisa mengetahui jumlah saldo langsung setelah melakukan transaksi perbankan.



Gambar 4.12. Cek Saldo

4.6. Pengujian Informasi

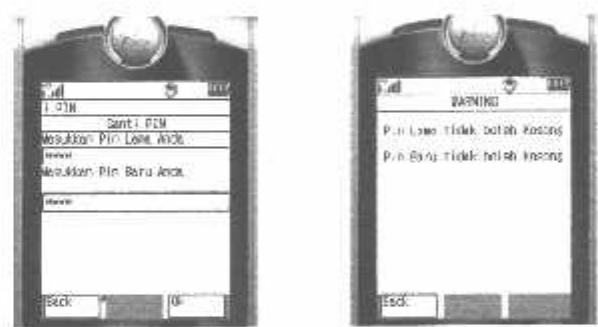
Pengujian Informasi ini adalah konten yang memberikan fasilitas kepada user untuk mendapatkan informasi tentang perbankan, informasi ini bersifat up to date yang mana selalu baru setiap harinya sesuai dengan perkembangan perbankan, selain informasi perkembangan perbankan bisa juga mengenai informasi tentang program – program baru dari pihak bank yang bersangkutan, jadi cakupan informasi yang di akses bias bermacam-macam.



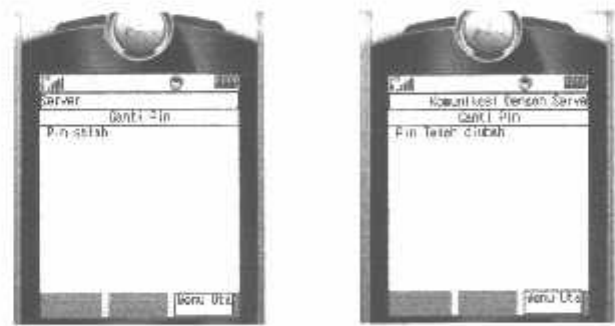
Gambar 4.13. Informasi dari Server

4.7. Pengujian Ganti Pin

Konten Penggantian PIN ini memungkinkan user untuk melakukan pergantian Pin secara berkala bisa meminimalisir tindak kejahatan sehingga PIN anda benar – benar terjaga, dalam pergantian Pin ini dilakukan pengisian terhadap Pin Lama dan Pin Baru, untuk menghindari pemalsuan user, oleh karena itu pengisian data Pin lama & Pin Baru di periksa dahulu apakah sudah benar – benar terisi Pin, setelah Pin lama dan Pin Baru telah terisi maka data tersebut dikirim ke server untuk memastikan user yang login saat itu memiliki Pin Lama yang di inputkan oleh user, jika pin lama salah maka permintaan pergantian Pin akan ditolak, begitupun sebaliknya jika Pin Lama benar atau sesuai dengan user yang sedang login maka permintaan di terima dan dipastikan permintaan pergantian Pin benar-benar dilakukan oleh user yang sebenarnya, bukan oleh orang lain.



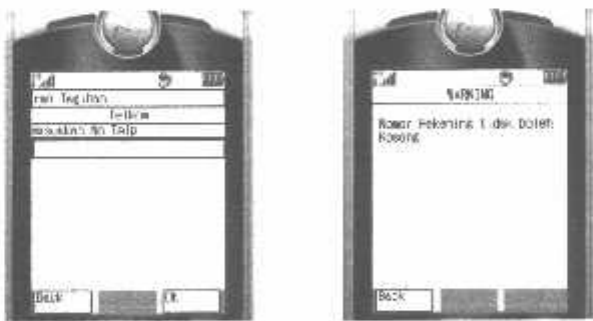
Gambar 4.14. Pergantian PIN & Konfirmasi Data



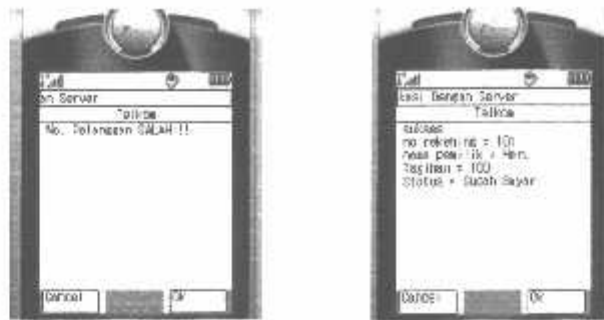
Gambar 4.15. PIN Salah & PIN telah di ubah

4.8. Pengujian Pembayaran tagihan

Konten Pembayaran tagihan memiliki fasilitas yang banyak sesuai dengan kerjasama antar divisi, sama seperti transaksi inputan nomor pelanggan yang dilakukan oleh user diperiksa terlebih dahulu untuk mengecek apakah user telah menginputkan nomor pelanggan untuk menghindari kekosongan data yang dikirim keserver, setelah melakukan pengisian No.Pelanggan maka dilakukan konfirmasi data ke server terlebih dahulu untuk mengetahui data dari nomor pelanggan yang diinputkan oleh user, dengan demikian kesalahan pembayaran rekening karena user salah memasukkan nomor pelanggan bisa dihindari, setelah user mengetahui data pemilik no pelanggan tersebut maka user bisa yakin melakukan pembayaran, jika sudah yakin maka data akan dikirim data ke server , dan server akan melakukan pengecekan pada status pelanggan apakah sudah membayar apa belum, jika belum membayar maka server melakukan pengecekan pada saldo user, jika saldo cukup maka server akan memproses jika tidak maka server tidak akan memproses, sedangkan jika status pelanggan sudah membayar maka proses pembayaran dihentikan untuk menghindari pembayaran dubel pada nomor pelanggan yang sama, jika nomor pelanggan salah atau tidak terdapat pada database maka user akan di peringatkan.



Gambar 4.16. Pembayaran & Konfirmasi



Gambar 4.17. Nomer Pelanggan Salah & Data Pelanggan



Gambar 4.18. Hasil Pegecekan Status



Gambar 4.19. Saldo Mencukupi & Saldo Tidak Mencukupi (Tagihan)

4.9. Pengujian Logout

Pada saat Logout maka user akan dihadapkan pada konfirmasi terlebih dahulu, baru setelah itu jika user setuju maka di lakukan penghapusan session pada server.



Gambar 4.20. Konfirmasi Logout & Logout

4.10. Pengujian Tarif & Data GPRS IM3

Pengujian tarif dan data GPRS di maksudkan untuk lebih jelas seberapa besar nilai dan waktu yang di gunakan oleh user untuk memanfaatkan mobile banking ini, dalam hal ini di pakai provider IM3 dimana pada IM3 tarif per kilobyte adalah Rp.1 , di sisi pelanggan ini sangat murah, tetapi di sisi bisnis antara bank dan provider ini sangat kecil nilai bisnisnya, masih jauh nilai bisnisnya dengan SMS Banking dimana tiap pengiriman Request dikenakan tariff Rp.350.

Tabel 4.1. Pengujian Waktu & Tarif

Kegiatan	Waktu	Rp	Bytes
Login	10 Detik	1.56	1560 Bytes
Isi Pulsa	3 Detik	1.36	1360 Bytes
Transfer	3 Detik	1.36	1360 Bytes
Cek Saldo	3 Detik	1.36	1360 Bytes
Pembayaran	3 Detik	1.36	1360 Bytes
Informasi	Tergantung paket data yang dikirim Server	-	Tergantung paket data yang dikirim Server
Ganti PIN	5 Detik	1.36	1360 Bytes
Logout	3 Detik	1.36	1360 Bytes

4.11. Pengujian Kinerja Aplikasi User

Pengujian kinerja aplikasi dilakukan untuk mengetahui sejauh mana keberhasilan Aplikasi dalam melakukan transaksi, dan kegiatan komunikasi

dengan web server, karena banyak hal yang mempengaruhi kinerja ini mulai dari Handphone user, jaringan GPRS dan juga web server pada Hosting yang kinerjanya masih jauh dibandingkan server Bank sesungguhnya, adapun beberapa kegiatan yang dilakukan pengujian demi mengetahui kinerja mobile banking ini, pengujian ini dilakukan dalam keadaan normal artinya sinyal dari provider full, adapun visualisasi dalam bentuk table untuk pengujian kinerja sebagai berikut :

Tabel 4.2. Kinerja Aplikasi

Kegiatan	Hasil
Login	Sukses
Isi Pulsa	Sukses
Transfer	Sukses
Cek Saldo	Sukses
Pembayaran	Sukses
Informasi	Sukses
Ganti PIN	Sukses
Logout	Sukses
Timer Security	Unsucces

BAB V

PENUTUP

5.1. Kesimpulan

1. Aplikasi mobile banking ini menggunakan pertukaran key untuk membuka aplikasi, dengan begitu form untuk transaksi dan administrasi hanya bisa di buka jika mendapatkan key dari server.
2. Data yang dikirim oleh user benar-benar terjaga karena alamat server yang di akses telah di sembunyikan dengan melakukan abstract pada variabel URL.

5.2. Saran

Kelemahan pada sistem ini yaitu sistem ini menggunakan server yang belum standart keamanan (SSL) namun jika akan di kembangkan untuk standart keamanan perbankan maka sangat mungkin dilakukan karena gateway komunikasi telah dibuka.

DAFTAR PUSTAKA

- [1] Developer Guide, *Java ME Developer Guide For Motorola OS*, Motorola, Inc 2008.
- [2] Technical Article, *Optimizing a Java ME Application Part 3: Canvas Performance Improvement*, Motorola, Inc 2007.
- [3] Shalahudin, M *Penrograman J2ME* Penerbit Informatika Bandung, 2006.
- [4] Gavin troy krugen, *Mobile Banking Technology Options*, TroyTyla, Inc
- [5] Wikipedia, 2008, Bank, URL: <http://id.wikipedia.org/bank>
- [6] Pemanfaatan “Ebanking” masih rendah, 6 juni 2002, www.sinarharapan.co.id
- [7] Sejarah_internet_indonesia: surat steven_haryanto_ke_bca_6_juni_2001, 6 agustus 2006, www.wikihost.org
- [8] Pengguna e-banking, waspadalah, 04 februari 2008, www.detikhot.com

LAMPIRAN



Formulir Perbaikan Skripsi

Dalam pelaksanaan ujian skripsi jenjang Strata satu (S-1) Jurusan Teknik Elektro konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk Mahasiswa :

Nama : HERRU PRASETIYA
NIM : 04.12.690
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Masa Bimbingan : 21 Juli 2008 s/d 21 Januari 2009
Judul Skripsi : Desain Aplikasi Mobile Banking Menggunakan JAVA

Tanggal	Uraian	Paraf
Penguji I 26 September 2008	1. Flowchart script Server. 2. Tabel database. 3. Spesifikasi Server & Database.	

Mengetahui,

Dosen Pembimbing I

Ir. F. Yudi Limpraptono, MT

NIP.Y. 1039500274

Dosen Pembimbing II

M. Ashar, ST, MT

NIP. P.1030500408

Dosen Penguji,

PENGUJI I

Dr. Cahyo Crysdian

NIP. Y. 103040412

PENGUJI II

Sotyohadi, ST, MT

NIP.Y. 1039700309



FORMULIR BIMBINGAN SKRIPSI

Nama : HERRU PRASETIYA
Nim : 04.12.690
Masa Bimbingan : 21 JULI 2008 s/d 21 JANUARI 2009
Judul Skripsi : DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA

No.	Tanggal	Uraian	Paraf Pembimbing
1.		Bab I	
2.		Bab II + Bab III	
3.		Demo	
4.	28/08/08	maahad seminar	
5.		Bab IV	
6.		BAB V	
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing

(Ir. F. Yudi Limpraptono, MT)
NIP.Y.1039500274

Form S-4B



FORMULIR BIMBINGAN SKRIPSI

Nama : HERRU PRASETIYA
Nim : 04.12.690
Masa Bimbingan : 21 JULI 2008 s/d 21 JANUARI 2009
Judul Skripsi : DESAIN APLIKASI MOBILE BANKING MENGGUNAKAN JAVA

No.	Tanggal	Uraian	Paraf Pembimbing
1.		Bab I , II (Revisi)	AS
2.		Bab III (Revisi)	AS
3.		Bab III. Rencana system	AS
4.		Bab IV Implementasi	AS
5.		Bab IV Hasil & Analisa	AS
6.		Bab IV Pengujian system	AS
7.		Bab IV System Logic	AS
8.		Bab V Kesimpulan	AS
9.		Mengetahui Seminar hasil	AS
0.		Korupsi	AS

Malang,
Dosen Pembimbing

(M Ashar, ST,MT)
NIP.Y.1030500408

Form S-4B

[illegible][illegible]

[illegible][illegible]

[illegible][illegible]

[illegible]

```

    }
    }

    private class() {
        @RequestMapping
        String result = imp.getMsg();
        return result;
    }

    @RequestMapping
    public String getResult() {
        try {
            return new ArrayList<ArgumentElement>().get(0).getResult();
        } finally {
            // 1. 调用连接HTTP
            if (conn != null) {
                conn.disconnect();
            }
        }
    }
}

```

```
class half_Takom {public: three {
    public Communist; // at Silver
    public void spellBooker(Can Vivaldi, balmering);
    balmering = cl
}
public: void half() {
    String str...
}
double = (re_Rayne_Takom.spellBooker());
```

```
try {
    tu = phrases_Kont_Talkern.hoer();
    str_Kont_Bayer_Telecom.satz(tu);
    beschLocation.satz(tu);
    System.out.println(tu);
    fms_Gesatz.satz(tu);
}
}
```

[illegible]

```
int size = 1;
StringBuffer tmp = new StringBuffer();
InputStreamReader input = new InputStreamReader(
    client.getInputStream());
char buffer[] = new char[size];

// Menampung pesan dari web server
while (!input.read(buffer, 0, buffer.length) == -1) {
    tmp.append(buffer);
    buffer = new char[size];
}

input.close();

// Kirimkan hasil
String result = tmp.toString();
return result;
}

} catch (Exception e) {
    throw new IOException(e.getMessage());
}

} finally {
    //5. Tutup koneksi !!!
}
```

4 points - full (1
correct),
3
}

```

class Test {
public:
    Test() {}
    Test(int i) {}
    Test(const Test& t) {}
    Test& operator=(const Test& t) {}
    ~Test() {}
};

int main() {
    Test t;
    Test t2(t);
    Test t3(t2);
    Test t4(t3);
    Test t5(t4);
    Test t6(t5);
    Test t7(t6);
    Test t8(t7);
    Test t9(t8);
    Test t10(t9);
    Test t11(t10);
    Test t12(t11);
    Test t13(t12);
    Test t14(t13);
    Test t15(t14);
    Test t16(t15);
    Test t17(t16);
    Test t18(t17);
    Test t19(t18);
    Test t20(t19);
    Test t21(t20);
    Test t22(t21);
    Test t23(t22);
    Test t24(t23);
    Test t25(t24);
    Test t26(t25);
    Test t27(t26);
    Test t28(t27);
    Test t29(t28);
    Test t30(t29);
    Test t31(t30);
    Test t32(t31);
    Test t33(t32);
    Test t34(t33);
    Test t35(t34);
    Test t36(t35);
    Test t37(t36);
    Test t38(t37);
    Test t39(t38);
    Test t40(t39);
    Test t41(t40);
    Test t42(t41);
    Test t43(t42);
    Test t44(t43);
    Test t45(t44);
    Test t46(t45);
    Test t47(t46);
    Test t48(t47);
    Test t49(t48);
    Test t50(t49);
    Test t51(t50);
    Test t52(t51);
    Test t53(t52);
    Test t54(t53);
    Test t55(t54);
    Test t56(t55);
    Test t57(t56);
    Test t58(t57);
    Test t59(t58);
    Test t60(t59);
    Test t61(t60);
    Test t62(t61);
    Test t63(t62);
    Test t64(t63);
    Test t65(t64);
    Test t66(t65);
    Test t67(t66);
    Test t68(t67);
    Test t69(t68);
    Test t70(t69);
    Test t71(t70);
    Test t72(t71);
    Test t73(t72);
    Test t74(t73);
    Test t75(t74);
    Test t76(t75);
    Test t77(t76);
    Test t78(t77);
    Test t79(t78);
    Test t80(t79);
    Test t81(t80);
    Test t82(t81);
    Test t83(t82);
    Test t84(t83);
    Test t85(t84);
    Test t86(t85);
    Test t87(t86);
    Test t88(t87);
    Test t89(t88);
    Test t90(t89);
    Test t91(t90);
    Test t92(t91);
    Test t93(t92);
    Test t94(t93);
    Test t95(t94);
    Test t96(t95);
    Test t97(t96);
    Test t98(t97);
    Test t99(t98);
    Test t100(t99);
    Test t101(t100);
    Test t102(t101);
    Test t103(t102);
    Test t104(t103);
    Test t105(t104);
    Test t106(t105);
    Test t107(t106);
    Test t108(t107);
    Test t109(t108);
    Test t110(t109);
    Test t111(t110);
    Test t112(t111);
    Test t113(t112);
    Test t114(t113);
    Test t115(t114);
    Test t116(t115);
    Test t117(t116);
    Test t118(t117);
    Test t119(t118);
    Test t120(t119);
    Test t121(t120);
    Test t122(t121);
    Test t123(t122);
    Test t124(t123);
    Test t125(t124);
    Test t126(t125);
    Test t127(t126);
    Test t128(t127);
    Test t129(t128);
    Test t130(t129);
    Test t131(t130);
    Test t132(t131);
    Test t133(t132);
    Test t134(t133);
    Test t135(t134);
    Test t136(t135);
    Test t137(t136);
    Test t138(t137);
    Test t139(t138);
    Test t140(t139);
    Test t141(t140);
    Test t142(t141);
    Test t143(t142);
    Test t144(t143);
    Test t145(t144);
    Test t146(t145);
    Test t147(t146);
    Test t148(t147);
    Test t149(t148);
    Test t150(t149);
    Test t151(t150);
    Test t152(t151);
    Test t153(t152);
    Test t154(t153);
    Test t155(t154);
    Test t156(t155);
    Test t157(t156);
    Test t158(t157);
    Test t159(t158);
    Test t160(t159);
    Test t161(t160);
    Test t162(t161);
    Test t163(t162);
    Test t164(t163);
    Test t165(t164);
    Test t166(t165);
    Test t167(t166);
    Test t168(t167);
    Test t169(t168);
    Test t170(t169);
    Test t171(t170);
    Test t172(t171);
    Test t173(t172);
    Test t174(t173);
    Test t175(t174);
    Test t176(t175);
    Test t177(t176);
    Test t178(t177);
    Test t179(t178);
    Test t180(t179);
    Test t181(t180);
    Test t182(t181);
    Test t183(t182);
    Test t184(t183);
    Test t185(t184);
    Test t186(t185);
    Test t187(t186);
    Test t188(t187);
    Test t189(t188);
    Test t190(t189);
    Test t191(t190);
    Test t192(t191);
    Test t193(t192);
    Test t194(t193);
    Test t195(t194);
    Test t196(t195);
    Test t197(t196);
    Test t198(t197);
    Test t199(t198);
    Test t200(t199);
    Test t201(t200);
    Test t202(t201);
    Test t203(t202);
    Test t204(t203);
    Test t205(t204);
    Test t206(t205);
    Test t207(t206);
    Test t208(t207);
    Test t209(t208);
    Test t210(t209);
    Test t211(t210);
    Test t212(t211);
    Test t213(t212);
    Test t214(t213);
    Test t215(t214);
    Test t216(t215);
    Test t217(t216);
    Test t218(t217);
    Test t219(t218);
    Test t220(t219);
    Test t221(t220);
    Test t222(t221);
    Test t223(t222);
    Test t224(t223);
    Test t225(t224);
    Test t226(t225);
    Test t227(t226);
    Test t228(t227);
    Test t229(t228);
    Test t230(t229);
    Test t231(t230);
    Test t232(t231);
    Test t233(t232);
    Test t234(t233);
    Test t235(t234);
    Test t236(t235);
    Test t237(t236);
    Test t238(t237);
    Test t239(t238);
    Test t240(t239);
    Test t241(t240);
    Test t242(t241);
    Test t243(t242);
    Test t244(t243);
    Test t245(t244);
    Test t246(t245);
    Test t247(t246);
    Test t248(t247);
    Test t249(t248);
    Test t250(t249);
    Test t251(t250);
    Test t252(t251);
    Test t253(t252);
    Test t254(t253);
    Test t255(t254);
    Test t256(t255);
    Test t257(t256);
    Test t258(t257);
    Test t259(t258);
    Test t260(t259);
    Test t261(t260);
    Test t262(t261);
    Test t263(t262);
    Test t264(t263);
    Test t265(t264);
    Test t266(t265);
    Test t267(t266);
    Test t268(t267);
    Test t269(t268);
    Test t270(t269);
    Test t271(t270);
    Test t272(t271);
    Test t273(t272);
    Test t274(t273);
    Test t275(t274);
    Test t276(t275);
    Test t277(t276);
    Test t278(t277);
    Test t279(t278);
    Test t280(t279);
    Test t281(t280);
    Test t282(t281);
    Test t283(t282);
    Test t284(t283);
    Test t285(t284);
    Test t286(t285);
    Test t287(t286);
    Test t288(t287);
    Test t289(t288);
    Test t290(t289);
    Test t291(t290);
    Test t292(t291);
    Test t293(t292);
    Test t294(t293);
    Test t295(t294);
    Test t296(t295);
    Test t297(t296);
    Test t298(t297);
    Test t299(t298);
    Test t300(t299);
    Test t301(t300);
    Test t302(t301);
    Test t303(t302);
    Test t304(t303);
    Test t305(t304);
    Test t306(t305);
    Test t307(t306);
    Test t308(t307);
    Test t309(t308);
    Test t310(t309);
    Test t311(t310);
    Test t312(t311);
    Test t313(t312);
    Test t314(t313);
    Test t315(t314);
    Test t316(t315);
    Test t317(t316);
    Test t318(t317);
    Test t319(t318);
    Test t320(t319);
    Test t321(t320);
    Test t322(t321);
    Test t323(t322);
    Test t324(t323);
    Test t325(t324);
    Test t326(t325);
    Test t327(t326);
    Test t328(t327);
    Test t329(t328);
    Test t330(t329);
    Test t331(t330);
    Test t332(t331);
    Test t333(t332);
    Test t334(t333);
    Test t335(t334);
    Test t336(t335);
    Test t337(t336);
    Test t338(t337);
    Test t339(t338);
    Test t340(t339);
    Test t341(t340);
    Test t342(t341);
    Test t343(t342);
    Test t344(t343);
    Test t345(t344);
    Test t346(t345);
    Test t347(t346);
    Test t348(t347);
    Test t349(t348);
    Test t350(t349);
    Test t351(t350);
    Test t352(t351);
    Test t353(t352);
    Test t354(t353);
    Test t355(t354);
```

```
by {
  try { process_Kunde_Hauptobjekt }
  catch {
    catch {
      catch {
        catch {
          catch {
            catch {
              catch {
                catch {
                  catch {
                    catch {
                      catch {
                        catch {
                          catch {
                            catch {
                              catch {
                                catch {
                                  catch {
                                    catch {
                                      catch {
                                        catch {
                                          catch {
                                            catch {
                                              catch {
                                                catch {
                                                  catch {
                                                    catch {
                                                      catch {
                                                        catch {
                                                          catch {
                                                            catch {
                                                              catch {
                                                                catch {
                                                                  catch {
                                                                    catch {
                                                                      catch {
                                                                        catch {
                                                                          catch {
                                                                            catch {
                                                                              catch {
                                                                                catch {
                                                                                  catch {
                                                                                    catch {
                                                                                      catch {
                                                                                        catch {
                                                                                          catch {
                                                                                           ...
                                                                                          }
                                                                                        }
                                                                                      }
                                                                                    }
                                                                                  }
                                                                                }
                                                                              }
                                                                            }
                                                                          }
                                                                        }
                                                                      }
                                                                    }
                                                                  }
                                                                }
                                                              }
                                                            }
                                                          }
                                                        }
                                                      }
                                                    }
                                                  }
                                                }
                                              }
                                            }
                                          }
                                        }
                                      }
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```
public String parse_Xml_MediaString(String input) throws IOException {
    HttpConnection conn = null;

```

```

by {
  N: lsbCountNonZeros
  String int = LPR_and_get(LsbCnt);
  count = (lbinCnt * factor) / (nonZeroCnt * N);
}

int size = 1;
StringBuffer tmp = new StringBuffer(1024);
InputSourceReader input = new InputSourceReader(new ByteArrayInputStream(""));
char[] buffer = new char[size];

// Main working paper can work server
while (input.read(buffer, 0, buffer.length) != -1) {
  // if accepted
  buffer = new char[size];
}

```

```
input:rcall;
A/D Kernballen theel nye
String %sulf- hup to 5ting0
cabin %sulf
```

```

1  patch: function (x) {
2    draw.rect(10, height - 10, width, height, color(255, 0, 0));
3  }
4  finally {
5    for (var i = 0; i < 10; i++) {
6      if (i % 2 == 0) {
7        color = color(0, 0, 255);
8      } else {
9        color = color(255, 0, 0);
10     }
11   }
12 }

```

```
class Kotlin extends Thread {
  private CommandListener listener
  public void setListener(CommandListener cl)
  listener = cl
}
public void run() {
  String s = ""
```

String repel = "M_Dejar_Matmosphere";

```
try {
    lu = p - seq_Next_Margin(margin);
    seq_Next_Deriv = Matrix(margin);
} catch(exception ex) {
    system.out.println(ex);
    PrintLog(ex.getMessage());
}
```

```
KCIPHER_AES_TRANSFDR = 0x0000000000000000
public String keyOf_Transfer(String inputurl, String inputurl2) throws IOException {
    HttpConnection conn = null;
```

```
try {
    //tindakan koneksi
    String url = URL;
    HttpURLConnection
    conn = (HttpURLConnection)connection.openConnection();

    int data = 1;

    String urlEmp = new String(buf.toString());
    InputstreamReader inStr = new InputstreamReader(conn.getInputStream());
    char[] buffer = new char[1024];

    //Membuang pesan dan web server
    while (inStr.read(buffer, 0, buffer.length)) != -1 {
        emp.append(buffer);
        buffer = new char[1024];
    }
}
```

```
input close;
05 kembalikan hasilnya
String result= tripInString;
return result;
```

```
1 catchException() {
2     this.newIllegalArgumentException("see GitHub page [0]");
3 }
```

A5. Futur bewerkstelligt HTTT

$\mathcal{B}$ (even in $\mathcal{M}$)
concluded).

class Kommantaa {
 transfer:refersTo Thread
 private CommentLustner beiter;
 public void setLustner (CommentLustner c) {
 beiter = c;
 }
}

Public and Private
Schools

```
across = buf_Hto_T.Transfer(getString());
string across = buf_Hto_T.Transfer().getString();
```

```
try {
    if (Fraggl[methodSign]) =
        u - Konf_Timestamp, normal);
    Elr_Konf_Timestamp ref(2019);
}
catch (Exception err) {
```

```
System.out.println(
    Fmt.format("Transfer amount: %d",
        amount));
}
```

[illegible]

```

194 # leave an empty
195 setting = URL_AblyURL()
196 conn = (p2p2d + "connect?connect=open1").
197
198 if not conn:
199     print("No conn")
200     return
201
202 # Send a GET request to the server
203 response = urllib2.urlopen(conn)
204
205 # Read the response
206 data = response.read()
207
208 # Print the response
209 print("Response: %s" % data)
210
211 # Close the connection
212 response.close()
213
214 # Print the response
215 print("Response: %s" % data)
216
217 # Close the connection
218 response.close()
219
220 # Print the response
221 print("Response: %s" % data)
222
223 # Close the connection
224 response.close()
225
226 # Print the response
227 print("Response: %s" % data)
228
229 # Close the connection
230 response.close()
231
232 # Print the response
233 print("Response: %s" % data)
234
235 # Close the connection
236 response.close()
237
238 # Print the response
239 print("Response: %s" % data)
240
241 # Close the connection
242 response.close()
243
244 # Print the response
245 print("Response: %s" % data)
246
247 # Close the connection
248 response.close()
249
250 # Print the response
251 print("Response: %s" % data)
252
253 # Close the connection
254 response.close()
255
256 # Print the response
257 print("Response: %s" % data)
258
259 # Close the connection
260 response.close()
261
262 # Print the response
263 print("Response: %s" % data)
264
265 # Close the connection
266 response.close()
267
268 # Print the response
269 print("Response: %s" % data)
270
271 # Close the connection
272 response.close()
273
274 # Print the response
275 print("Response: %s" % data)
276
277 # Close the connection
278 response.close()
279
280 # Print the response
281 print("Response: %s" % data)
282
283 # Close the connection
284 response.close()
285
286 # Print the response
287 print("Response: %s" % data)
288
289 # Close the connection
290 response.close()
291
292 # Print the response
293 print("Response: %s" % data)
294
295 # Close the connection
296 response.close()
297
298 # Print the response
299 print("Response: %s" % data)
300
301 # Close the connection
302 response.close()
303
304 # Print the response
305 print("Response: %s" % data)
306
307 # Close the connection
308 response.close()
309
310 # Print the response
311 print("Response: %s" % data)
312
313 # Close the connection
314 response.close()
315
316 # Print the response
317 print("Response: %s" % data)
318
319 # Close the connection
320 response.close()
321
322 # Print the response
323 print("Response: %s" % data)
324
325 # Close the connection
326 response.close()
327
328 # Print the response
329 print("Response: %s" % data)
330
331 # Close the connection
332 response.close()
333
334 # Print the response
335 print("Response: %s" % data)
336
337 # Close the connection
338 response.close()
339
340 # Print the response
341 print("Response: %s" % data)
342
343 # Close the connection
344 response.close()
345
346 # Print the response
347 print("Response: %s" % data)
348
349 # Close the connection
350 response.close()
351
352 # Print the response
353 print("Response: %s" % data)
354
355 # Close the connection
356 response.close()
357
358 # Print the response
359 print("Response: %s" % data)
360
361 # Close the connection
362 response.close()
363
364 # Print the response
365 print("Response: %s" % data)
366
367 # Close the connection
368 response.close()
369
370 # Print the response
371 print("Response: %s" % data)
372
373 # Close the connection
374 response.close()
375
376 # Print the response
377 print("Response: %s" % data)
378
379 # Close the connection
380 response.close()
381
382 # Print the response
383 print("Response: %s" % data)
384
385 # Close the connection
386 response.close()
387
388 # Print the response
389 print("Response: %s" % data)
390
391 # Close the connection
392 response.close()
393
394 # Print the response
395 print("Response: %s" % data)
396
397 # Close the connection
398 response.close()
399
400 # Print the response
401 print("Response: %s" % data)
402
403 # Close the connection
404 response.close()
405
406 # Print the response
407 print("Response: %s" % data)
408
409 # Close the connection
410 response.close()
411
412 # Print the response
413 print("Response: %s" % data)
414
415 # Close the connection
416 response.close()
417
418 # Print the response
419 print("Response: %s" % data)
420
421 # Close the connection
422 response.close()
423
424 # Print the response
425 print("Response: %s" % data)
426
427 # Close the connection
428 response.close()
429
430 # Print the response
431 print("Response: %s" % data)
432
433 # Close the connection
434 response.close()
435
436 # Print the response
437 print("Response: %s" % data)
438
439 # Close the connection
440 response.close()
441
442 # Print the response
443 print("Response: %s" % data)
444
445 # Close the connection
446 response.close()
447
448 # Print the response
449 print("Response: %s" % data)
450
451 # Close the connection
452 response.close()
453
454 # Print the response
455 print("Response: %s" % data)
456
457 # Close the connection
458 response.close()
459
460 # Print the response
461 print("Response: %s" % data)
462
463 # Close the connection
464 response.close()
465
466 # Print the response
467 print("Response: %s" % data)
468
469 # Close the connection
470 response.close()
471
472 # Print the response
473 print("Response: %s" % data)
474
475 # Close the connection
476 response.close()
477
478 # Print the response
479 print("Response: %s" % data)
480
481 # Close the connection
482 response.close()
483
484 # Print the response
485 print("Response: %s" % data)
486
487 # Close the connection
488 response.close()
489
490 # Print the response
491 print("Response: %s" % data)
492
493 # Close the connection
494 response.close()
495
496 # Print the response
497 print("Response: %s" % data)
498
499 # Close the connection
500 response.close()
501
502 # Print the response
503 print("Response: %s" % data)
504
505 # Close the connection
506 response.close()
507
508 # Print the response
509 print("Response: %s" % data)
510
511 # Close the connection
512 response.close()
513
514 # Print the response
515 print("Response: %s" % data)
516
517 # Close the connection
518 response.close()
519
520 # Print the response
521 print("Response: %s" % data)
522
523 # Close the connection
524 response.close()
525
526 # Print the response
527 print("Response: %s" % data)
528
529 # Close the connection
530 response.close()
531
532 # Print the response
533 print("Response: %s" % data)
534
535 # Close the connection
536 response.close()
537
538 # Print the response
539 print("Response: %s" % data)
540
541 # Close the connection
542 response.close()
543
544 # Print the response
545 print("Response: %s" % data)
546
547 # Close the connection
548 response.close()
549
550 # Print the response
551 print("Response: %s" % data)
552
553 # Close the connection
554 response.close()
555
556 # Print the response
557 print("Response: %s" % data)
558
559 # Close the connection
560 response.close()
561
562 # Print the response
563 print("Response: %s" % data)
564
565 # Close the connection
566 response.close()
567
568 # Print the response
569 print("Response: %s" % data)
570
571 # Close the connection
572 response.close()
573
574 # Print the response
575 print("Response: %s" % data)
576
577 # Close the connection
578 response.close()
579
580 # Print the response
581 print("Response: %s" % data)
582
583 # Close the connection
584 response.close()
585
586 # Print the response
587 print("Response: %s" % data)
588
589 # Close the connection
590 response.close()
591
592 # Print the response
593 print("Response: %s" % data)
594
595 # Close the connection
596 response.close()
597
598 # Print the response
599 print("Response: %s" % data)
600
601 # Close the connection
602 response.close()
603
604 # Print the response
605 print("Response: %s" % data)
606
607 # Close the connection
608 response.close()
609
610 # Print the response
611 print("Response: %s" % data)
612
613 # Close the connection
614 response.close()
615
616 # Print the response
617 print("Response: %s" % data)
618
619 # Close the connection
620 response.close()
621
622 # Print the response
623 print("Response: %s" % data)
624
625 # Close the connection
626 response.close()
627
628 # Print the response
629 print("Response: %s" % data)
630
631 # Close the connection
632 response.close()
633
634 # Print the response
635 print("Response: %s" % data)
636
637 # Close the connection
638 response.close()
639
640 # Print the response
641 print("Response: %s" % data)
642
643 # Close the connection
644 response.close()
645
646 # Print the response
647 print("Response: %s" % data)
648
649 # Close the connection
650 response.close()
651
652 # Print the response
653 print("Response: %s" % data)
654
655 # Close the connection
656 response.close()
657
658 # Print the response
659 print("Response: %s" % data)
660
661 # Close the connection
662 response.close()
663
664 # Print the response
665 print("Response: %s" % data)
666
667 # Close the connection
668 response.close()
669
670 # Print the response
671 print("Response: %s" % data)
672
673 # Close the connection
674 response.close()
675
676 # Print the response
677 print("Response: %s" % data)
678
679 # Close the connection
680 response.close()
681
682 # Print the response
683 print("Response: %s" % data)
684
685 # Close the connection
686 response.close()
687
688 # Print the response
689 print("Response: %s" % data)
690
691 # Close the connection
692 response.close()
693
694 # Print the response
695 print("Response: %s" % data)
696
697 # Close the connection
698 response.close()
699
700 # Print the response
701 print("Response: %s" % data)
702
703 # Close the connection
704 response.close()
705
706 # Print the response
707 print("Response: %s" % data)
708
709 # Close the connection
710 response.close()
711
712 # Print the response
713 print("Response: %s" % data)
714
715 # Close the connection
716 response.close()
717
718 # Print the response
719 print("Response: %s" % data)
720
721 # Close the connection
722 response.close()
723
724 # Print the response
725 print("Response: %s" % data)
726
727 # Close the connection
728 response.close()
729
730 # Print the response
731 print("Response: %s" % data)
732
733 # Close the connection
734 response.close()
735
736 # Print the response
737 print("
```

```
if (isStreaming) {
    readFromWebServer(
        (input, readBuffer, 0, buffer.length) -> {
            drop append(buffer);
            buffer = new char[256];
        }
    );
}
```

```
input classf;
AP: Kembalikan hasilnya
String result= tmp.classing();
return result;
```

```

1 catch(Exception ex)
2 {
3     throw new ArgumentException("exception not good", ex);
4 }
5 finally
6 {
7     // Do something HTTP
8 }
9 }
10 }
11 }
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

```

[illegible]

String input = "C:\Users\gustavo\Documents\Di...

```

    }
    // -long zxcvbnq);
    str_Hash_Ads=0;
}
//Exemption are:
System.out.println("
from_Hash_Ads.append(""+name+"");

```

IBM SA 1

```

    do {
        public void run() {
            String s="";
            while (true) {
                if (s.length() < 10) {
                    s = s + "a";
                } else {
                    s = s.substring(1);
                }
            }
        }
    }
}

```

```

try {
    rs = Htmg_zzconnect();
    Str_Html_3.connect(rs);
}
catch(exception ex) {
    System.out.println("
    Fm_Html_3.appendSignalNetwork");
}

```

SI PULSA MENTARI

```
class PubSub_Mem {
public:
    PubSub_Mem() {}
    ~PubSub_Mem() {}
    void AddSubscriber(const string &subscriber) {
        subscribers.push_back(subscriber);
    }
    void RemoveSubscriber(const string &subscriber) {
        subscribers.erase(remove(subscribers.begin(), subscribers.end(), subscriber), subscribers.end());
    }
    void Publish(const string &message) {
        for (const string &subscriber : subscribers) {
            cout << subscriber << " received message: " << message << endl;
        }
    }
};
```

$$\text{Strong input} = C_{\text{g_Membangatsizing}}/C_{\text{g_MembangatsizingIndex}}$$

```

14 # (Using_ZooFormat).
15 fit_H2O_MermaidTest()

```

```
cat(&exception, err)
system("cd ..");
free_Host_Memory(approx("yagel korolov"))
}
```

ISFLUEA INC. INFORMATION AVAILABLE

```
case Pulse_M5 extend t' need (
  probe CommrClusterItem::c
  pulse void sedlshener (CommrClusterItem::c) {
    [b] = 0;
  }
}
```

public weed control
\$200,000

Strong input = Cg_Mgstr(Cg_MgSelectedIndex);

```
try {
    rs = fitting_zzox(input),
    Gu_Howl_M3.startEnd(rs)
}
```

```
catchException err {
  System.out.println(err)
  File last_MW_append("signal_monitoring")
}
```

```

    }
    ..... IS_PULSA_AS .....
}

```

```

class PULSA_AS extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
    String input = Cg_Lib.getting(Cg_Lib.getSocketIn());
}
}

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_As.set(rs);
} catch (Exception ex) {
    System.out.println("
    PULSA_AS: error! gagal koneksi!");
}
}
}

```

```

..... IS_PULSA_SMPA .....

```

```

class PULSA_SMPA extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
}
}

```

```

String input = Cg_Smpa.getting(Cg_Smpa.getSocketIn());

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_Smpa.set(rs);
} catch (Exception ex) {
    System.out.println("
    PULSA_SMPA: error! gagal koneksi!");
}
}
}

```

```

..... IS_PULSA_EMPO .....

```

```

class PULSA_EMPO extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
}
}

```

```

String input = Cg_Empo.getting(Cg_Empo.getSocketIn());

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_Empo.set(rs);
} catch (Exception ex) {
}
}

```

```

    System.out.println("
    PULSA_EMPO: error! gagal koneksi!");
}
}

```

```

..... IS_PULSA_PFBAG .....
class PULSA_PFBAG extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
    String input = Cg_Pfbag.getting(Cg_Pfbag.getSocketIn());
}
}

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_Pfbag.set(rs);
} catch (Exception ex) {
    System.out.println("
    PULSA_PFBAG: error! gagal koneksi!");
}
}
}

```

```

..... IS_PULSA_FIEK .....

```

```

class PULSA_FIEK extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
}
}

```

```

String input = Cg_Fiek.getting(Cg_Fiek.getSocketIn());

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_Fiek.set(rs);
} catch (Exception ex) {
    System.out.println("
    PULSA_FIEK: error! gagal koneksi!");
}
}
}

```

```

..... IS_PULSA_SMPA .....

```

```

class PULSA_SMPA extends Thread {
private CommandListener ls;
public void setListener (CommandListener cl) {
    listener = cl;
}
public void run() {
    String rs="";
}
}

```

```

String input = Cg_Smpa.getting(Cg_Smpa.getSocketIn());

```

```

try {
    rs = Hilung.zzz(input);
    Str_Hasil_Smpa.set(rs);
} catch (Exception ex) {
    System.out.println("
    PULSA_SMPA: error! gagal koneksi!");
}
}
}

```

[illegible][illegible]

```

class Input_Editable {
public:
    Input_Editable() {}
    Input_Editable(const string &str) {
        str = str;
    }
    string input = "C++ is a great language!";
};

```

```

try {
    P2 = Hlsreq.get(0).part1;
    S27 = Hlsreq.get(0).part2;
}
catch(E:Exception) {}
System.out.println("
    P1m_Hlsreq_Ess speed: " + P2 + "
    P1m_Hlsreq_Ess speed: " + S27 + "
    ");
}
}

```

```
TRANSA EN PROBABILITIES OF OCCURRENCE
public String hitURL(String inputURL) throws IOException {
    HttpURLConnection conn = null;

```

```

try {
    // bekommen kenne ich
    String url = URL.addGetParameter(
        conn -> (HttpURLConnection) conn.getOutputStream(),
        "lat=" +
            String.valueOf(lat) + "&lon=" +
            String.valueOf(lon));
    InputStream inputStream = conn.getInputStream();
    String json = new String(inputStream.readAllBytes());
    // Merkspruch: genau der web server
    // antwortet (input.readAllBytes() liefert json) = {}
    try {
        JSONObject obj = new JSONObject(json);
        // double -> new class (float);
    } catch (JSONException e) {}
} catch (IOException e) {}
}

```

AS: Kembalikan hatinya
Gifting: kembalikan impi-impinya
return: kembalikan

```

1  f.write('accepting\n')
2  # now have things in front of us, get the scope()
3  # and f
4  finally:
5      # do cleanup housework
6      f.close()
7
8  # accept is null
9  # conn closed.
10

```

1. *Chlamydomonas reinhardtii* (green alga)
 2. *Chlamydomonas reinhardtii* (green alga)
 3. *Chlamydomonas reinhardtii* (green alga)
 4. *Chlamydomonas reinhardtii* (green alga)
 5. *Chlamydomonas reinhardtii* (green alga)
 6. *Chlamydomonas reinhardtii* (green alga)
 7. *Chlamydomonas reinhardtii* (green alga)
 8. *Chlamydomonas reinhardtii* (green alga)
 9. *Chlamydomonas reinhardtii* (green alga)
 10. *Chlamydomonas reinhardtii* (green alga)

Offenbar ist $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^*)$.

```
try:
    if P.empty() method signature
    re = Turing.Tester(t, p, data)
    out = Pool.ThreadSafeWriter()

except Exception as e:
    System.out.println(e);
    from .Haskell.Tester import * as HaskellTester
```

```
public String getPropertyByName(String name) throws IOException {
    if (isConnected() == null)

```

```
try {
    A labson kernel!
    String str = "HELLO, world!"
    return "v" + str + "connected successfully"
}
```

```
int size = 1;
StringBuffer sbuff = new StringBuffer(size);
InputStreamReader input = new InputStreamReader(new FileInputStream(
    configFile), "UTF-8");
```

^a Mutating process that will set new while if input variable is 0 (buffer length is 0) respectively.

```
} else if ($?) {  
    throw new ArgumentException("getServer()"  
        , "only {  
            or 'http://www.foo.com' or  
            'ftp://www.foo.com' allowed."  
        )  
}
```

[illegible][illegible]

