

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Perancangan merupakan suatu rancangan untuk menentukan perancangan dalam pembuatan desain/rancangan yang akan dibuat pada suatu penelitian yang terdapat pada judul tersebut. Perancangan ataupun perancangan merupakan salah satu sarana manajemen untuk mencapai tujuan yang telah ditetapkan karena itu setiap tingkat manajemen dalam organisasi sangat membutuhkan aktivitas perancangan dan tujuan dari perancangan adalah salah satu sarana manajemen untuk mencapai tujuan yang telah ditetapkan karena itu setiap tingkat manajemen dalam organisasi sangat membutuhkan aktivitas perancangan.

Perancangan merupakan desain awal dalam sebuah pembuatan penelitian yang akan dibuat dalam konsep atau rancangan yang ada tersebut. Perancangan digolongkan sebagai fakta yang *objective* kebenarannya bahwa pemikiran yang rasional itu tidak atas hayalan belaka tetapi suatu perhitungan berdasarkan data yang *objective*. Walau perancangan mengandung unsur dugaan/pemikiran namun harus didasarkan pada suatu standar yang terukur. Perancangan adalah sebagai tahap persiapan / tindakan pendahuluan untuk melaksanakan kegiatan dengan memperhatikan penyimpangan yang mungkin terjadi.

II.2. Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

II.3 Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Kamus berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (*etimologi*) sesuatu perkataan dan juga contoh penggunaan bagi suatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus.

Kata kamus diserap dari bahasa Arab *qamus*, dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *okeanos* yang berarti samudra. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan *khazanah* yang membuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya.

II.3.1. Jenis-Jenis Kamus

Kamus dapat di tulis dalam satu atau lebih bahasa, dan berdasarkan penggunaan bahasanya kamus dapat dibagi menjadi 3 jenis yaitu :

1. Kamus Ekabahasa

Kamus ini hanya menggunakan satu bahasa. Kata-kata yang dijelaskan dan penjelasannya adalah terdiri daripada bahasa yang sama. Kamus ini mempunyai perbedaan yang jelas dengan kamus dwibahasa kerana penyusunan dibuat berasaskan pembuktian data korpus. Ini bermaksud definisi makna ke atas kata-kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandung kata-kata berhubungan. Contoh kamus ekabahasa ialah kamus Besar Bahasa Indonesia (Indonesia) dan Kamus Dewan di (Malaysia).

2. Kamus Dwibahasa

Kamus ini menggunakan dua bahasa, yakni kata masukan daripada bahasa yang dikamuskan diberi padanan atau pemberian takrifnya dan arti dengan menggunakan bahasa yang lain. Contohnya: Kamus Inggris-Indonesia, Kamus Dwibahasa *Oxford Fajar* (Inggris-Melayu; Melayu-Inggris).

3. Kamus Aneka Bahasa

Kamus ini sekurang-kurangnya menggunakan tiga bahasa atau lebih. Misalnya, kata Bahasa Melayu Bahasa Inggris dan Bahasa Mandarin secara serentak.

Kamus dapat dibagi dalam berbagai ukuran, karena kamus diterbitkan dengan tujuan memenuhi keperluan golongan tertentu. Contohnya, golongan pelajar sekolah memerlukan kamus berukuran kecil untuk memudahkan mereka

membawa kamus ke sekolah. Secara umum kamus dapat dibagi kepada 3 (tiga) jenis ukuran, yaitu:

1. Kamus Mini,

Pada zaman sekarang sebenarnya susah untuk menjumpai kamus mini. Kamus mini juga dikenali sebagai kamus saku karena dapat disimpan dalam saku. Tebal kamus mini tidak lebih dari 2 cm.

2. Kamus Kecil,

Kamus kecil merupakan kamus berukuran kecil yang biasa dijumpai di kalangan masyarakat. Kamus kecil merupakan kamus yang mudah dibawa namun ukurannya jauh lebih besar dari kamus mini dan tidak dapat disimpan di dalam saku. Contoh kamus kecil salah satunya, kamus Dwibahasa Oxford Fajar (Inggris-Melayu;Melayu-Inggris).

3. Kamus Besar,Kamus besar memuatkan segala leksikal yang terdapat dalam satu bahasa. Setiap perkataan akan dijelaskan maksud dan artinya secara lengkap. Umumnya ukuran kamus besar dapat dikatakan cukup besar dan tidak sesuai untuk dibawa ke sana sini. Contohnya Kamus Besar Bahasa Indonesia.

Kamus bukan hanya digunakan untuk menerjemahkan kata dalam suatu bahasa ke bahasa lain, tetapi kamus juga digunakan untuk menerjemahkan makna kata yang merupakan istilah ataupun peribahasa. Kamus tersebut disebut kamus istimewa. Kamus istimewa merujuk kepada kamus yang mempunyai fungsi yang khusus. Contohnya:

1. Kamus Istilah,

Kamus ini berisi istilah-istilah khusus dalam bidang tertentu. Fungsinya adalah untuk kegunaan ilmiah. Contohnya ialah Kamus Istilah Fiqh

2. Kamus Etimologi,

Kamus yang menerangkan asal usul sesuatu perkataan dan maksud asalnya.

3. Kamus Tesaurus (perkataan searti),

Kamus yang menerangkan maksud sesuatu perkataan dengan memberikan kata-kata searti (sinonim) dan dapat juga kata-kata yang berlawanan arti (antonim). Kamus ini adalah untuk membantu para penulis untuk meragamkan penggunaan diksi. Contohnya, Tesaurus Bahasa Indonesia

4. Kamus Peribahasa/Simpulan Bahasa,

Kamus yang menerangkan maksud sesuatu peribahasa/simpulan bahasa. Selain daripada digunakan sebagai rujukan, kamus ini juga sesuai untuk dibaca dengan tujuan keindahan.

5. Kamus Kata Nama Khas,

Kamus yang hanya menyimpan kata nama khas seperti nama tempat, nama tokoh, dan juga nama bagi institusi. Tujuannya adalah untuk menyediakan rujukan bagi nama-nama ini.

6. Kamus Terjemahan,

Kamus yang menyediakan kata arti bahasa asing untuk satu bahasa sasaran. Kegunaannya adalah untuk membantu para penerjemah.

7. Kamus Kolokasi,

Kamus yang menerangkan pandangan kata, contohnya kata ‘terdiri’ yang selalu berpandangan dengan ‘dari atau ‘atas’.

Kamus mobile yang akan dirancang termasuk jenis kamus terjemahan karena kamus ini berfungsi untuk menterjemahkan kebahasa sasaran, yakni bahasa Arab sebagai masukan dari bahasa yang dikamuskan dan diterjemahkan atau artinya menggunakan bahasa Indonesia.

II.4. Bahasa

Bahasa adalah penggunaan kode yang merupakan gabungan fonem sehingga membentuk kata dengan aturan sintaks untuk membentuk kalimat yang memiliki arti. Bahasa memiliki berbagai definisi. Definisi bahasa adalah sebagai berikut:

1. Suatu sistem untuk mewakili benda, tindakan, gagasan dan keadaan.
2. Suatu peralatan yang digunakan untuk menyampaikan konsep riil mereka ke dalam pikiran orang lain.
3. Suatu kesatuan sistem makna.
4. Suatu kode yang digunakan oleh pakar linguistik untuk membedakan antara bentuk dan makna.
5. Suatu ucapan yang menepati tata bahasa yang telah ditetapkan (contoh: perkataan, kalimat, dan lain-lain.).
6. Suatu sistem tuturan yang akan dapat dipahami oleh masyarakat linguistik.

II.4.1. Bahasa Arab

Bahasa Arab adalah salah satu bahasa Semitik Tengah, yang termasuk dalam rumpun bahasa Semitik dan berkerabat dengan bahasa Ibrani dan bahasa-bahasa Neo Arami. Bahasa Arab memiliki lebih banyak penutur daripada bahasa-bahasa lainnya dalam rumpun bahasa Semitik. Ia dituturkan oleh lebih dari 280 juta orang sebagai bahasa pertama, yang mana sebagian besar tinggal di Timur Tengah dan Afrika Utara. Bahasa ini adalah bahasa resmi dari 25 negara, dan merupakan bahasa peribadatan dalam agama Islam karena merupakan bahasa yang dipakai oleh Al-Qur'an. Berdasarkan penyebaran geografisnya, bahasa Arab percakapan memiliki banyak variasi (dialek), beberapa dialeknya bahkan tidak dapat saling mengerti satu sama lain.

Bahasa Arab telah memberi banyak kosakata kepada bahasa lain dari dunia Islam, sama seperti peranan Latin kepada kebanyakan bahasa Eropa. Semasa Abad Pertengahan bahasa Arab juga merupakan alat utama budaya, terutamanya dalam sains, matematik dan filsafah, yang menyebabkan banyak bahasa Eropa turut meminjam banyak kosakata dari bahasa Arab.

II.4.2. Bahasa Indonesia

Bahasa Indonesia adalah bahasa resmi Republik Indonesia dan bahasa persatuan bangsa Indonesia. Bahasa Indonesia diresmikan penggunaannya setelah Proklamasi Kemerdekaan Indonesia, tepatnya sehari sesudahnya, bersamaan dengan mulai berlakunya konstitusi. Di Timor Leste, bahasa Indonesia berstatus sebagai bahasa kerja.

Dari sudut pandang linguistik, bahasa Indonesia adalah salah satu dari banyak ragam bahasa Melayu. Dasar yang dipakai adalah bahasa Melayu Riau (wilayah Kepulauan Riau sekarang) dari abad ke-19. Dalam perkembangannya ia mengalami perubahan akibat penggunaannya sebagai bahasa kerja di lingkungan administrasi kolonial dan berbagai proses pembakuan sejak awal abad ke-20. Penamaan "Bahasa Indonesia" diawali sejak dicanangkannya Sumpah Pemuda, 28 Oktober 1928, untuk menghindari kesan "imperialisme bahasa" apabila nama bahasa Melayu tetap digunakan. Proses ini menyebabkan berbedanya Bahasa Indonesia saat ini dari varian bahasa Melayu yang digunakan di Riau maupun Semenanjung Malaya. Hingga saat ini, Bahasa Indonesia merupakan bahasa yang hidup, yang terus menghasilkan kata-kata baru, baik melalui penciptaan maupun penyerapan dari bahasa daerah dan bahasa asing.

Meskipun dipahami dan dituturkan oleh lebih dari 90% warga Indonesia, Bahasa Indonesia bukanlah bahasa ibu bagi kebanyakan penuturnya. Sebagian besar warga Indonesia menggunakan salah satu dari 748 bahasa yang ada di Indonesia sebagai bahasa ibu. Penutur Bahasa Indonesia kerap kali menggunakan versi sehari-hari (kolokial) dan/atau mencampuradukkan dengan dialek Melayu lainnya atau bahasa ibunya. Meskipun demikian, Bahasa Indonesia digunakan sangat luas di perguruan-perguruan, di media massa, sastra, perangkat lunak, surat-menyurat resmi, dan berbagai forum publik lainnya, sehingga dapatlah dikatakan bahwa Bahasa Indonesia digunakan oleh semua warga Indonesia.

II.5. *Mobile Device*

Mobile Device merupakan suatu alat yang digunakan oleh *pemakai* untuk meminta informasi yang dibutuhkan, dimana informasi dapat diberikan dalam bentuk suara, gambar, dan *text*. Informasi yang diinginkan dapat dicari melalui fasilitas untuk mengakses *internet* seperti GPRS atau *wireless*. Pada umumnya perangkat *mobile* atau *Mobile device* lebih praktis karena bersifat mudah dibawa (*portable*) dari pada perangkat teknologi lainnya ([http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=27: layanan-berbasis-lokasi &catid=25:industri&Itemid=14](http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=27:layanan-berbasis-lokasi&catid=25:industri&Itemid=14)).

Meningkatnya pemakaian peranti *mobile* (*mobile device*) telah merevolusi kegiatan-kegiatan yang bersifat tradisional menjadi lebih sederhana dan mudah dengan penggunaan perangkat *mobile*. Mobilitas yang tinggi tidak menjadi penghalang lagi, karena saat ini peranti *mobile* sudah dapat mengakses *server* atau pusat data. Peranti *mobile* sekarang tidak hanya berfungsi sebagai pencatat jadwal dan buku alamat. Fungsi peranti *mobile* sudah berkembang pesat dan idealnya siap mengganti dokumen berbasis kertas. (Tri Mardiono, 2006, chap.1)

Vendor-vendor peranti *mobile* telah menanam teknologi penangkap data (*data acquisition*) ke dalam produknya. Hal ini menciptakan peningkatan produktivitas, proses pelaporan yang lebih cepat, dan pengurangan biaya operasional. Secara umum aplikasi peranti *mobile* terbagi atas:

1. *Personal Information Managenent* (PIM)

Menyediakan fungsi kalender, buku alamat, jadwal, memo, pengirim *email* dan tugas yang harus dilakukan.

2. Dokumen

Menyediakan fungsi *word processing* dan *spread sheet*. Fungsi yang lebih maju antara lain : akses *web*, *e-book*, multimedia, dan presentasi.

3. Aplikasi *Mobile Business* (*mBusiness*)

Menyediakan fungsi penangkapan, pemrosesan, dan pengiriman data. Komunikasi yang umum antara *server* dan aplikasi adalah melalui sistem *messaging* seperti SMS dan sebagainya. Namun sekarang aplikasi *business* sudah dapat terhubung langsung dengan *server* melalui HTTP secara *prorietary* atau melalui *web service*. Dengan J2ME peranti *mobile* yang menggunakan CDC bisa melakukan koneksi berbobot melalui RME, JDBC dan sebagainya.

4. *Games* dan hiburan

Menyediakan fungsi permainan, musik, video dan sebagainya. J2ME menyediakan *library-library* untuk membangun aplikasi *games* dan hiburan.
(Tri Mardiono, 2006, chap.1)

II.5.1 Jenis – Jenis *Mobile Device*

Mobile device dapat dibagi berdasarkan jumlah kegunaannya menjadi dua jenis yaitu:

1. *Single Purpose*

Mobile device dengan *single purpose* digunakan untuk satu tujuan saja.
contoh: *navigation box* yang terdapat pada mobil atau truk pengangkut barang.

2. *Multi Purpose*

Mobile device dengan *multi purpose* dapat digunakan untuk mengakses berbagai macam informasi yang diinginkan, contoh : *Smart phone*, *handphone*, PDA, Laptop.

(<http://www.ittelkom.ac.id/library/index.php?view=article&catid=25:industr&id=237:layanan-berbasis-lokasi&format=pdf>).

Aplikasi terjemahan yang akan penulis rancang menggunakan perangkat *mobile* jenis *multi purpose* yaitu *handphone*, dimana *handphone* yang digunakan harus mendukung aplikasi Java™.

II.6. Sekilas Tentang Bahasa Pemrograman Java™

Java™ merupakan bahasa pemrograman yang dikembangkan dengan menggunakan bahasa C, sehingga pengembang (*programmer*) C tidak mengalami kesulitan beralih ke Java™. Java™ diciptakan oleh James Gosling dan Patrick Naughton dalam suatu proyek di *Sun Microsystem* sekitar tahun 1991. Pada mulanya ingin diberi nama OAK yang berasal dari nama pohon yang terdapat pada kantor James Gosling, namun karena kata OAK telah ada pada *Sun Microsystem*, maka diberi nama Java™ yang terinspirasi dari minum kopi. (Yuniar Supardi Ir, 2008, chap.1)

Java™ merupakan bahasa pemrograman *multiplatform*, sehingga banyak segmen yang memakainya. Bahasa Java™ meliputi pemrograman *dekstop*, pemrograman *database*, bahasa pemrograman *mobile*, dan lain-lain. Java™ juga

portable, karena semua sistem operasi seperti *Windows*, *Linux*, *Unix*, dan lain-lain dapat menjalankan *Java*TM. *Java*TM dibagi menjadi tiga edisi (*Platform*) yaitu :

1. J2SE (*Java*TM 2 *Standard Edition*)

Merupakan edisi standard (*basis*) dari *Java*TM. J2SE lebih difokuskan pada pemrograman *Dekstop* dan *Applet* (aplikasi yang didapat dijalankan di *browser web* seperti *Internet Explorer*, *Firefox Mozilla*, *Opera*, dan lain-lain).

2. J2EE (*Java*TM 2 *Enterprise Edition*)

Merupakan edisi perluasan dari J2SE (Superset dari J2SE), aplikasi yang dibuat dengan edisi ini untuk aplikasi berskala besar (*Enterprise*), seperti pemrograman memakai *database* dan di atur oleh *server*.

3. J2ME (*Java*TM 2 *Micro Edition*)

J2ME merupakan edisi khusus dari *Java*TM dan subset dari edisi J2SE. Edisi ini untuk pemrograman dengan peralatan-peralatan kecil atau terbatas seperti *PDA*, *handphone*, *pager*, dan lain-lain. (Yuniar Supardi Ir, 2008. chap.1).

II.6.1. J2ME (*Java*TM 2 *Micro Edition*)

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 *bandwith* jaringan yang rendah. Dengan perkembangan perangkat *mobile* konsumen dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, *Java*TM menyediakan suatu lingkungan yang *portable* untuk mengembangkan dan menjalankan aplikasi pada

perangkat ini (Yuniar Supardi Ir, 2008, chap.1). Teknologi *Java*TM mempunyai banyak keunggulan, diantaranya :

1. *Multiplatform*

Aplikasi J2ME dapat berjalan dibanyak *platform* yang didalamnya terdapat JVM (*Java*TM *Virtual Machine*), beberapa *platform* yang tersedia *installer mobile* JVM antara lain: *Windows*, *Symbian*, dan *Embedded Linux*.

2. *Robust*

Kode-kode *Java*TM adalah kode-kode yang *robust* karena *virtual machine* mengatur keamanan proses eksekusi aplikasi. JVM menyediakan *garbage collector* yang bertugas mencegah kebocoran *memory*.

3. Terintegrasi dengan baik

J2ME bisa terhubung dengan *back end* J2EE *server* dan *web service* dengan mudah, karena J2ME menyediakan *librar-library* API RMI dan *web service*.

4. Berorientasi objek

*Java*TM merupakan salah satu bahasa pemrograman yang murni berorientasi objek. Hal ini mempermudah dan mempercepat pengembangan sistem yang dikembangkan dengan metode analisa dan desain berorientasi objek. (Tri Mardiono, 2006, chap.1).

Configuration merupakan *Java*TM *library* minimum dan kapabilitas yang dipunyai oleh para pengembang J2ME, yang maksudnya sebuah *mobile device* dengan kemampuan *Java*TM akan dioptimalkan untuk menjadi sesuai (Shalhuddin M & Rossa, 2008, chap.1). Ada 2 (dua) konfigurasi yang didefinisikan dalam

J2ME yaitu: CLDC (*Connected Limited Device Configuration*) untuk perangkat kecil dan CDC (*Connected Device Configuration*) untuk perangkat yang lebih besar.

CLDC (*Connected Limited Device Configuration*) adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *Library* dan *API* yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon seluler, *pager*, dan PDA (Shalhuddin M & Rossa, 2008, chap.1). Perangkat tersebut dibatasi dengan keterbatasan *memory*, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *Package*, kelas dan sebagian fungsi *JavaTM Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*).

CDC (*Connected Device Configuration*) adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada *JavaTM* yang memiliki standarisasi. CDC terdiri dari *virtual machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *Source code* yang menyediakan sambungan dengan macam-macam *Platform* (Shalhuddin M & Rossa, 2008, chap.1).

Profile berbeda dengan *configuration*, *profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Dalam J2ME terdapat 2 (dua) buah *profile* yaitu: MIDP dan *Foundation Profile*. MIDP (*Mobile Information Device Profile*) adalah spesifikasi untuk sebuah *profile* J2ME. MIDP memiliki lapisan di atas CLDC,

API tambahan untuk daur hidup tambahan aplikasi, antar muka, jaringan dan penyimpan persisten.

Tabel II.1. Arsitektur J2ME

Profile (MIDP)	
Configuration (CLDC/CDC)	Kumpulan <i>Library</i>
	KVM/JVM
Sistem Operasi	

Sumber : Shalhuddin M & Rossa, 2008, chap.1

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

Untuk memudahkan penulis mengetikkan program *Java*TM maka penulis menggunakan JCreator. JCreator merupakan perangkat lunak editor untuk *Java*TM. Tampilan menu utama JCreator dapat dilihat pada gambar II.1 berikut :

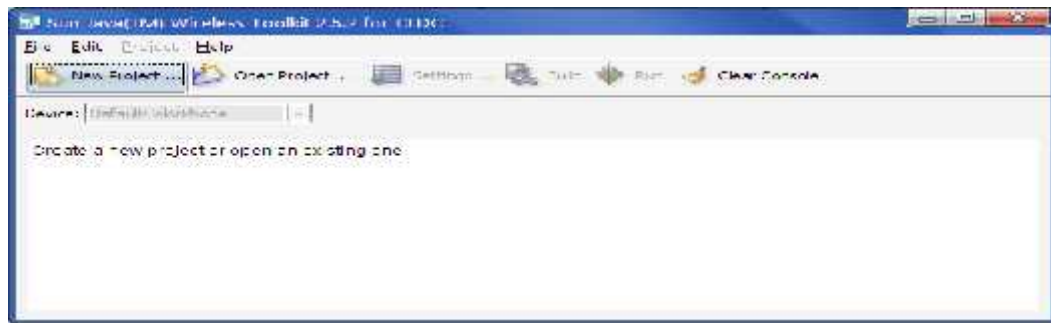


Gambar II.1. Menu Utama Jcreator

Sumber: Yuniar Supardi Ir.,2008, chap.1

Untuk menjalankan program *Java* yang telah diketikan pada JCreator, penulis menggunakan Sun *Java* Wireless Toolkit 2.5.2 for CLDC sebagai emulatorenya. Sun *Java* WTK (Wireless Toolkit) merupakan perangkat lunak atau tool emulator mensimulasikan kerja handphone, sehingga pada waktu membuat program handphone, pemogram tidak perlu mencoba langsung atau mengkoneksikan ke internet.

Berikut ini tampilan menu utama Sun *Java* Wireless Toolkit 2.5.2 for CLDC :



Gambar II.2. Menu Utama Sun Java Wireless Toolkit 2.5.2 for CLDC

Sumber: Rangsang Purnama, 2010, chap.1

Aplikasi akan diuji dengan menggunakan emulator Sun Java(TM) Wireless Toolkit 2.5.2 for CLDC dan S80 3rd Edition SDK FP 1 for MIDP. Hal ini bertujuan untuk memperoleh data-data yang akurat mengenai aplikasi baik pada tahap pengkodean maupun uji coba sebelum digunakan pada telepon genggam yang sebenarnya.



Gambar II.3. Emulator +5550000

Sumber: Rangsang Purnama, 2010, chap.1

II.7 RMS (*Record Management System*)

Pemrograman J2ME tidak mengenal *database*, untuk menyimpan data dikenal dengan nama RMS (*Record Management System*) yang merupakan mekanisme penyimpanan berbentuk *record*. Dalam pemrograman MIDP terdapat 3 ruang penyimpanan data yang dapat digunakan, yaitu: *Volatile* (sementara) RAM, ruang persisten (tetap), dan penyimpanan secara *remote*.

Data yang disimpan didalam RAM bersifat *volatile* (sementara), jika peralatan dimatikan atau MIDlet ditutup (dimatikan) data akan hilang. Pengaturan RAM sudah diatur oleh JVM (*Java Virtual Machine*) dengan teknologi *garbage collector* tanpa intervensi *programmer* (Yuniar Supardi Ir, 2008, chap.2).

Pada MIDlet yang terhubung dengan suatu jaringan internet, dapat mengakses atau menyimpan data pada ruang penyimpanan *remote* misalnya *server database*. (Yuniar Supardi Ir, 2008, chap.4).

Record pada RMS disimpan sebagai *array* dari *byte* yang cara kerjanya berdasarkan *record* (baris data). RMS memiliki orientasi *record* basis data yang sederhana sehingga tidak mengenal *field* (kolom data) seperti *database* pada umumnya. Dalam RMS tidak bisa mengambil data per *field* per *record* seperti yang biasa dilakukan pada *database* yang umum, sehingga perlu dipetakan dahulu datanya

Paket RMS terdapat pada *Javax.microedition.rms* yang memiliki beberapa *interface* dan satu kelas. Satu-satunya kelas yang digunakan untuk memanipulasi data adalah kelas *RecordStore*. Kelas dan *Interface* yang terdapat pada paket RMS adalah :

1. Kelas *RecordStore*.

Kelas ini digunakan untuk membuka, membaca, menulis, mengubah, menghapus dan menutup data RMS.

2. Interface *RecordEnumeration*.

Interface ini digunakan untuk menelusuri data RMS akan tetapi tidak dapat digunakan untuk menambah, mengubah atau menghapus data RMS. Fungsi lain dari *interface* ini adalah untuk menyaring dan mengurutkan data.

3. Interface *RecordComparator*.

Interface ini berguna untuk membandingkan dua *record* dengan metode *compare()* terimplementasi. Hasil dari perbandingan ini akan berupa konstanta integer statis: *PRECEDES* (parameter pertama bernilai lebih kecil), *EQUIVALENT* (kedua parameter bernilai sama) dan *FOLLOW* (parameter pertama bernilai lebih besar). Fungsi lain dari *interface* ini adalah menampilkan data yangurut pada *interface RecordEnumeration*.

4. Interface *RecordFilter*.

Interface ini digunakan untuk menguji *record* dengan metode *matches()* terimplementasi yang akan mengembalikan nilai boolean: *true* (bila lulus uji) dan *false* (bila tidak lulus uji). Kegunaan lain dari *interface* ini yaitu untuk menyaring data pada *interface RecordEnumeration*.

II.8. Alat Bantu Perancangan Sistem

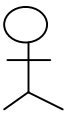
Adapun alat bantu yang digunakan dalam perancangan atau pembangunan sistem yang digunakan dalam penelitian umumnya berupa gambaran atau diagram yang tertera di bawah ini.

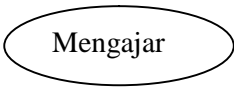
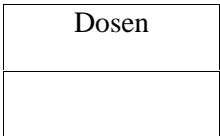
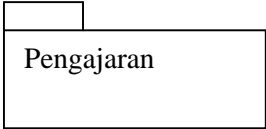
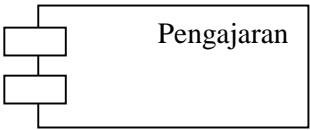

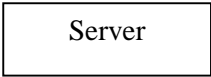
II.8.1 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah ‘bahasa’ atau pemodelan untuk sistem atau perangkat lunak yang berparadikma ‘berorientasi objek’. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Adi Nugroho, 2010, chap.1)

Objek pada umumnya merupakan potongan *state* yang dapat didefinisikan dengan perilaku yang terdefinisi dengan baik, yang dapat dipanggil oleh berapa bagian-bagian lain sistem atau perangkat lunak yang membutuhkannya. Relasi yang terjadi antar pengklasifikasi (*classifier*) sering dijumpai dalam berbagai bentuk asosiasi (*assosiation*), generalisasi (*generalization*) serta sebagai jenis kebergantungan (*dependency*). Berikut tabel pengklasifikasi (*classifier*) :

Tabel II.2. Pengklasifikasi (*Clasifier*)

Notasi	Pengklasifikasi	Kegunaan
	<i>Actor</i>	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem atau perangkat

Mahasiswa		lunak) yang berinteraksi dengan sistem yang dikembangkan.
	<i>Use Case</i>	Mengembangkan fungsional yang dimiliki sistem
	Kelas (<i>class</i>)	Menggambarkan konsep dasar pemodelan sistem
	Subsistem (SubSystem)	Menggambarkan paket spesifikasi serta implementasi
	Komponen (Component)	Menggambarkan bagian-bagian fisik sistem atau perangkat lunak yang dikembangkan
 Antarmuka Pengajar- Mahasiswa	Antarmuka (<i>Interface</i>)	Menggambarkan antarmuka pengiriman pesan (<i>massege</i>) antar perangkat lunak
	Simpul (<i>node</i>)	Menggambarkan sumber daya komputasional yang digunakan oleh sistem




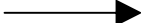

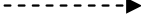
Sumber : Adi Nugroho, 2010, chap.1

Kaidah-kaidah yang dapat digunakan untuk melakukan *transformasi* dari diagram-diagram UML ke bahasa-bahasa pemrograman berorientasi tertentu,

kelas-kelas implementasi sesungguhnya menangkap bentuk langsung suatu kelas yang diperlukan oleh bahasa pemrograman berorientasi objek yang dipilih.

Berikut ini tabel relasi-relasi dalam UML :

Tabel II.3. Relasi-Relasi dalam UML


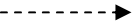
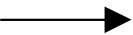
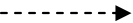
Relasi	Fungsi	Notasi
Asosiasi (<i>Association</i>)	Mendeklarasikan hubungan antar <i>instance</i> suatu kelas	
Kebergantungan (<i>dependency</i>)	Relasi antar dua elemen model	
Aliran (<i>flow</i>)	Relasi antar dua versi suatu objek	
Generalisasi (<i>Generalization</i>)	Relasi antar pengklasifikasi yang memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasi yang lebih spesifik, digunakan dalam struktur pewarisan	
Relasasi (<i>relazation</i>)	Relasi antar spesifikasi dan implementasi	
Penggunaan (<i>usage</i>)	Situasi dimana salah satu elemen membutuhkan elemen yang lain agar dapat berfungsi dengan baik	

Sumber : Adi Nugroho, 2010, chap.1

II.8.2 Use Case

Use case sesungguhnya merupakan unit koheren dari fungsionalitas dari sistem atau perangkat lunak yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada diluar sistem. Kegunaan *use case* sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyingkapkan struktur *internal* sistem atau perangkat lunak yang sedang dikembangkan. Masing-masing *use case* merepresentasikan fungsionalitas mandiri dimana eksekusinya dapat digabungkan dengan eksekusi *use case* lainnya.

Tabel II.4. Relasi Dalam Use Case

Relasi	Fungsi	Notasi
Asosiasi (<i>Association</i>)	Lintasan komunikasi antara <i>actor</i> dengan <i>use case</i>	
<i>Extend</i>	Penambahan suatu perilaku ke suatu <i>use case</i> dasar	
Generalisasi <i>Use Case</i>	Menggambarkan hubungan antar <i>use case</i> yang bersifat umum dengan <i>use case</i> yang bersifat lebih spesifik	
<i>Include</i>	Penambahan perilaku ke suatu <i>use case</i> dasar yang secara eksplisit	

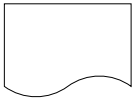
	mendeklarasikan penambahan tersebut	
--	-------------------------------------	--

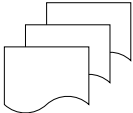
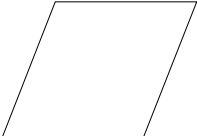


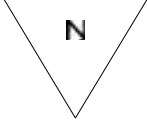
Sumber : Adi Nugroho, 2010, chap.1


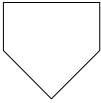
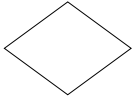

II.8.3 Sytem Flowchart

System flowchart adalah perangkat diagram grafik yang menyimpan dan mengkomunikasikan aliran data media dan prosedur proses informasi yang diperlukan dalam Pemrograman Microsoft. Hal ini dilakukan dengan menggunakan berbagai simbol yang dihubungkan dengan panah-panah untuk menunjukkan kelanjutan aktivitas proses informasi. Sistem *flowchart* tertentu berfungsi penting sebagai media dan *hardware* yang digunakan dan proses yang berhubungan dengan Pemrograman Microsoft. Semua itu mewakili model grafis dari Pemrograman Microsoft fisik yang diperlukan atau diajukan. Sistem ini banyak dipakai untuk menghubungkan struktur menyeluruh dan aliran sistem ke pengguna akhir karena sistem ini dapat menawarkan tampilan fisik yang berperan penting pada keterkaitan *hardware* dan data media. Walaupun begitu, beberapa kasus, sistem tersebut dapat digantikan dengan diagram aliran data untuk digunakan oleh analis sistem profesional, dan dengan grafik presentasi untuk berkomunikasi dengan pengguna akhir.

Tabel II.5. Simbol – Simbol Bagan Alir (*Flowchart*)

Simbol	Nama	Keterangan
	Dokumen	Dokumen tersebut dapat dipersiapkan dengan tulisan tangan

		atau dicatat dengan komputer
	Beberapa tembusan dari satu dokumen	Digambarkan dengan cara menumpuk simbol dokumen dan mencetidak nomor dokumen di bagian depan sudut kanan atas
	<i>Input/Output</i>	Fungsi <i>input</i> dan <i>output</i> apapun didalam bagan alir program, juga digunakan dalam mewakili jurnal dan buku besar dalam bagan alir dokumen
	Pemrosesan dengan komputer	Biasanya menghasilkan perubahan atas data atau informasi.
	Proses manual	Pelaksanaan pemrosesan yang dilakukan secara manual
	<i>File</i>	<i>File</i> dokumen secara manual disimpan. Huruf yang ditulis di dalam simbol menunjukkan urutan pengaturan <i>file</i> secara N = numeris, A = Alfabetis, D = Tanggal

	Arus dokumen atau proses	Arah pemrosesan atau arus dokumen
	<i>Off-page connector</i>	Suatu penanda masuk dari atau keluar ke halaman lain
	Keputusan	Langkah pengambilan keputusan
	<i>Terminal</i>	Titik awal,akhir atau pemberhentian dalam suatu proses.

Sumber : Jugiyanto HM., 2001, chap.4