BAB II
LANDASAN TEORI

2.1 Teori Dasar Sistem

2.1.1 Perancangan

Menurut Verdi Yasin, (2012) Perancangan adalah proses mendefinisikan suatu model atau rancangan perangkat lunak dengan menggunakan teknik dan prinsip tertentu sedemikian hingga model atau rancangan tersebut dapat diwujudkan menjadi perangkat lunak.

Menurut Verdi Yasin, (2012) tujuan dilakukannya perancangan oleh seorang *desaigner* sistem (*software engineer*) adalah :

a. Menjelaskan fungsionalitas masing-masing komponen.

b. Menentukan antarmuka komponen.

c. Mendekomposisi sistem (perangkat lunak) menjadi komponen komponennya (data, antarmuka, prosedur, arsitektur)

d. Menentukan mekanisme komunikasi antar komponen.

e. Menentukan relasi antar tabel.

f. Sebagai gambaran yang menunjukkan mekanisme dan relasi antar komponen perangkat lunak yaitu relasi antarmuka pemakai ke prosedur/*script* untuk meminta sebuah data yang diinginkan pengguna serta bagaimana sebuah prosedur mengakses tabel data agar dapat ditampilkan sesuai dengan permintaan pemakai pada antarmuka pemakai.

2.1.2 Sistem

Menurut Tata Sutabri (2012) Sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.Model umum sebuah sistem adalah *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran.Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa

dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut "supra sistem".

2. Batasan Sistem

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkung luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem.Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara.Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung sistem

   Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari subsistem akan menjadi masukan untuk subsistem

lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem

   Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan *(maintenance input)* dan sinyal *(signal input).* Contoh, di dalam suatu unit sistem komputer, "Program" adalah *maintenance* input yang digunakan untuk mengoperasikan komputernya dan "Data" adalah signal *input* untuk diolah menjadi informasi.

6. Keluaran Sistem*(Output)*

   Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain.

7. Pengolahan Sistem *(Process)*

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem *(Objective)*

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*.Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya.Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

2.1.3 Informasi

Menurut Tata Sutabri (2012) Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya.Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang.Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.

2.1.4 Sistem Informasi

Menurut Tata Sutabri (2012) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan- laporan yang diperlukan.

2.2 Teori Pengembangan Sistem

2.2.1 *Unified Modeling Language* (UML)

Menurut Verdi Yasin, (2012) mendefinisikan *Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebauh standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Menurut Verdi Yasin (2012) tipe-tipe Diagram UMLadalah sebagai berikut.

a. Use Case Diagram

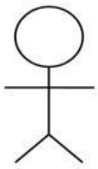
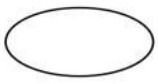
Use Case Diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuahsistem, yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case*empresentasikan sebuah interaksi antara *actor* dan sistem.

Komponen-komponen relasi dalam use case antara lain :

1. *Assocciation,* Menghubungkan *link* antar element.
2. *Generaliziont,* Disebut juga *interitance* (pewaris), sebuah elemen dapatmerupakan spesialis dari elemen lainnya.
3. *Depedency,* Sebuah element bergantung dalam beberapa cara ke elemenlainnya.
4. *Aggregation,* Bentuk *association* dimana sebuah elemen berisi elemenlainnya.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *caserelationship*. Penjelasan simbol pada tabel 2.1

Tabel 2.1 Simbol *Use Case*

Simbol	Keterangan
	**Aktor** : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	***Use case*** : perungkat tertinggi dari fungsionalitas yang dimiliki sistem.
	Association *:* adalah relasi antara actor dan *use case.*
	Generalisasi *:* untuk memperlihatkan struktur pewaris yang terjadi.
	Include **:** menspesifikasikan bahwa *use case* sumber secara *eksplisit*
	***Collaboration :*** interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemen nya (sinergi)
	***Extend :*** menspesifikasikan bahwa *use case* target memperluar perilaku dari *use case* sumber pada suatu titik yang diberikan

b. Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa *flowchart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

1. *Activity*

   Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.
2. *Transition*

   Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity.*
3. *Decision*

   Notasi yang menandakan kontrol cabang aliran berdasarkan *decision point.*
4. *Synchonization bars*

   Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan.

Komponen-komponen dari *activity diagram* dapat dilihat pada tabel 2.2 berikut :

Tabel 2.2 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

c. *Sequence Diagram*

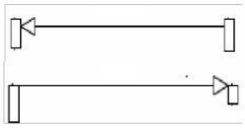
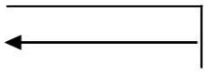
Sequence diagram menggambarkan kolaborasi dinamis antara sejumalah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Dalam *sequence* diagram terdapat 2 simbol yaitu :

1. Actor, untuk menggambarkan pengguna sistem.
2. *Lifeline,* untuk menggambarkan kelas dan objek.

Komponen-komponen dari *Sequence Diagram* dapat dilihat pada tabel 2.3 berikut:

Tabel 2.3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Messege to self*	Menggambarkan pesan yang menuju dirinya sendiri
4		*Return Message*	Menggambarkan pengembalian dari pemanggilan prosedur

d. Class Diagram

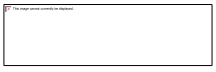
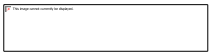
Class diagram menggambarkan struktur data dan deskripsi *class, package,* dan objek beserta hubungan satu sama lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. *Class* memiliki 3 area pokok antara lain :

1. Nama
2. Atribut
3. Metode

Class menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

Komponen-komponen dari *Class Diagram* dapat dilihat pada tabel 2.4 berikut:

Tabel 2.4 Simbol *Class Diagram*

Nama Komponen	Keterangan	Simbol
Class	*Class* adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan property/atribut *class*. Bagian akhir mendefinisikan *method-method* dari sebuah *class*.	Nama *Class* + atribut + atribut + atribut + *method* + *method*
Association	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 *class* dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship*. (Contoh: *One-to-one*, *one-to-many*, *many-to-many*).	
Composition	Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
Dependency	Kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik.	

Aggregation	*Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi	

2.2.2 **Metode *Rational Unified Process* (RUP)**

Menurut Rosa A.S dan M.Shalahuddin (2013) *Unified Process* atau di kenaljuga proses iteratif dan *incremental* merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan *incremental* (bertahap dengan proses menaik). Iteratif bisa dilakukan di dalam setiap tahap, atau iteratif tahap proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah menaik) di mana setiap interasi akan memperbaiki interasi berikutnya. Salah satu *unified process* yang terkenal adalah RUP (*rational unified process*). RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih di arahkan berdasarkan pengguna kasus (*use case driven*). RUP merupakan proses perangkat lunak dengan pendefinisian yang baik (*well defined*) dan perstrukturan yang baik (*well struktured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *rational sofware* yang diakuisisi oleh IBM di bulan Februari 2003

Proses pengulangan/iteraktif pada RUP secara global dapat dilihat pada gambar 2.1 berikut :

Gambar 2.1 Proses Iteratif RUP

a. Kelebihan RUP

Pendekatan iteratif/pengulangan dari RUP dapat mengokomodir beberapa kelemahan pengembangan perangkat lunak tanpa menggunakan konsep pengulangan.

1. RUP mengakomodasikan perubahan kebutuhan perangkat lunak. Kebutuhan untuk mengubah dan menambah fitur karena perubahan teknologi atau keinginan pelanggan pelanggan (*customer*) merupakan salah satu kendala yang sering dialami pengembangan perangkat lunak yang berimbas pada terlambatnya waktu penyelesaian perangkat lunak. Keterlambatan ini dapat menyebabkan ketidakpuasan di sisi pelanggan (*customer*) dan pengembang menjadi frustasi.
2. Integrasi bukanlah sebuah proses besar dan cepat (*big bag*) di akhir proyek. Menepatkan integrasi dibagian akhir proyek biasanya memakan waktu proyek yang cukup banyak, bisa mencapai 40 persen

dari waktu proyek. Untuk mencegah hal ini maka pendekatan secara iteratif (pengulahan) dapat memecah proyek menjadi bagian iterasi kecil yang di akhiri dengan integrasi kecil yang nantinya digabungkan menjadi integrasi besar.

3. Risiko biasanya ditemukan atau di alamatkan selama pada proses intergrasi awal. Pendekatan intergrasi pada RUP mengurangi risiko pada iterasi awal dimana saat semua komponen diuji.
4. Manajemen berarti membuat perubahan taktik pada produk. Taktik produk misalnya pengembangan dengan waktu singkat akan menghasilkan produk dengan fungsi yang terbatas akan dapat cepat digunakan oleh *user* sehingga memperkenalkan produk lebih cepat kemasyarakat dibanding produk kompetitor lain yang masih sedang dikembangkan.
5. Mendukung fasilitas penggunaan kembali

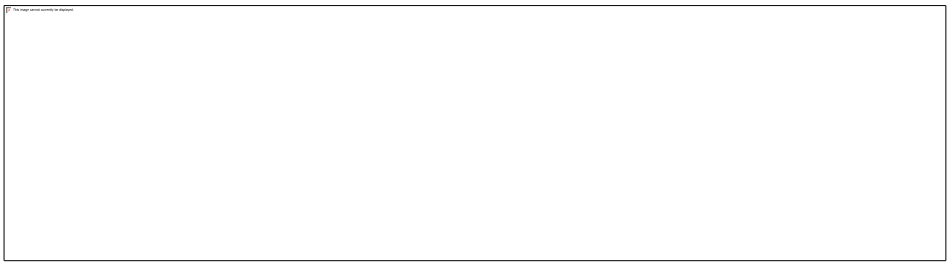
   Lebih mudah untuk mengidentifikasi bagian umum yang sering digunakan dalam aplikasi jika diimplementasikan secara iterasi daripada mengidentifikasi pada saat perencanaan saja.
6. Kecacatan dapat ditemukan dapat diperbaiki pada beberapa iterasi menghasilkan arsitektur yang baik dan aplikasi berkualitas tinggi Kecacatan dideteksi lebih baik mulai pada iterasi awal daripada pada tahap pengujian
7. Lebih baik menggunakan “anggota proyek” dibanding susunan secara seri pada tim proyek.
8. Anggota tim belajar selama proyek berjalan.

   Anggota proyek memiliki peluang belajar dari kesalahan selama siklus pengembangan perangkat lunak dan memperbaiki kesalahan pada interasi berikutnya.Kesalahan yang terjadi pada iterasi sebelumnya dapat diperbaiki pada iterasi berikutnya.
9. Pengembangan perangkat lunak dapat diperbaiki seiring proses pengembangan perangkat lunak

Setiap akhir proses iterasi tidak hanya ada peninjauan mengenai target produk tetapi juga peninjauan pada kesalahan yang terjadi pada iterasi sebelumnya agar dapat diperbaiki pada iterasi berikutnya.

b. Fase RUP

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif.Berikut ini adalah gambar 2.2 alur hidup RUP.



Gambar 2.2 Alur Hidup RUP

Berikut ini penjelasan untuk setiap fase pada RUP.

1. *Inception* (Permulaan)

Tahap ini lebih pada pemodelan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requiretments*).

Berikut adalah tahap yang dibutuhkan pada tahap ini:

a. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhannya, risiko dan lain sebagainya).

b. Membangun kasus bisnis yang dibutuhkan.

Hasil yang diharapkan dari tahap ini adalah memenuhi *lifecycle objective milestone* (batasan/tonggak objektif dari siklus) dengan kriteria berikut

a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal.
b. Kebutuhan dimengerti dengan pasti (dapat dibuktikan) dan sejalan dengan kasus primer yang dibutuhkan

c. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, risiko dan proses pengembangan.
d. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan
e. Membangun garis dasar dengan membandingkan perencanaan aktual dengan perencanaan yang direncanakan

   Jika pada akhirnya tahap ini target yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali setelah dirancang ulang agar kriteria yang diinginkan dapat dicapai. Batas/tonggak objektif digunakan untuk mendeteksi apakah sebuah kebutuhan akan sistem dapat di implementasi atau tidak.

2. *Elaboration* (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem.Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak.Mendeteksi risiko yang mungkin terjadi dari arsitektur yang dibuat.Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem *(prototype).*Hasil yang diharapkan dari tahap ini adalah memenuhi *lifecycle architecturemilestone* (batas/tonggak arsitektur dari siklus) dengan kriteria berikut:

a. Model kasus yang digunakan *(use case)* dimana kasus dan aktor yang terlihat telah didefinisikan dari sebagian besar kasus harus dikembangkan. Model *use case* harus 80 persen lengkap dibuat.
b. Deskripsi dari arsitektur perangkat lunak dari proses pengembangan sistem perangkat lunak telah dibuat.
c. Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case.*
d. Kasus bisnis dan proses bisnis dan daftar risiko yang sudah mengalami perbaikan (revisi) telah dibuat.
e. Rencana pengembangan untuk seluruh proyek telah dibuat.
f. Purwarupa *(prototype)* yang dapat di demonstrasikan untuk mengurangi setiap risiko teknis yang didefinisikan.

Jika pada akhir tahap ini dapat yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali. Batas/tonggak arsitektur digunakan untuk mendeteksi apakah sebuah kebutuhan akan sistem dapat diimplementasikan atau tidak melalui pembuatan arsitektur.

3. *Construction* (Konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem.Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program.Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Intial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*.Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan u*ser*.

Produk perangkat lunak juga disesuaikan dengan kebutuhan yang didefinisikan pada tahap *inception.*Jika semua kriteria objektif terpenuhi maka dianggap sudah memenuhi *Products Release Milestone* (batas/tonggak peluncuran produk) dan pengembangan perangkat lunak selesai dilakukan.Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap.Ke empat fase pada rup dijalankan secara berurutan dan iteratif dimana setiap iterasi dapat digunakan untuk memperbaiki iterasi berikutnya.

2.3 Android

Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa

melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

2.3.1 Karakteristik Android

Android memiliki empat karakteristik sebagai berikut :

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (third-party application). Semua apliksi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dpat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

2.3.2 Android Software Development Kit (SDK)

Android SDK adalah tool API (application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang release oleh Google. Saat ini di sediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemograman java (Nasruddin Safaat H 2012).

2.3.3 Android Virtual Device (AVD)

AVD yang merupakan emulator untuk menjalankan program aplikasi android yang kita buat, AVD ini nantinya yang kita jadikan sebagai tempat test dan menjalankan aplikasi android yang kita buat, AVD berjalan di virtual Manchine (Nasruddin Safaat H 2012)

2.4 Eclipse

Menurut Nasruddin Safaat h (Pemrograman aplikasi mobeli smartphone dan tablet PC berbasis android 2012) Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent).

2.5 Notepad++

Menurut Bunafit Nugroho. “Notepad ++ adalah sebuahsoftware bawaan windows sebagai editor dasar”. Notepad ++ merupakan software yang dapat membantu kita

membuat HTML pada Web Satu halyang harus diperhatikan dalam membuat HTML menggunakan notepad ++yaitu menentukan Type file saat penyimpanan

2.6 Aplikasi Mobile

Aplikasi *mobile* adalah *software* yang didesain untuk dapat dijalankan pada *smartphones,* komputer tablet, dan perangkat *mobile* lainnya. Pada awalnya, aplikasi *mobile* ditawarkan untuk produk yang dapat menunjang produktivitas pengguna secara umum seperti *email*, kalender, kontak, dan informasi cuaca. Akan tetapi, permintaan publik dan pertumbuhan pengembang aplikasi *mobile* yang meningkat pesat, memperluas cakupan aplikasi *mobile* seperti *mobile games*, aplikasi *banking*, dan majalah *online*.

2.7 Smartphone

Telepon cerdas (smartphone) adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. Bagi beberapa pendapat, telepon pintar merupakan telepon yang bekerja menggunakan sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, telepon cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku elektronik (e-book) dan lainnya.

Dengan kata lain, telepon cerdas merupakan sebuah telepon genggam yang mempunyai kemampuan sebuah komputer. (Atika, 2013)

Dalam penelitian ini smartphone adalah media yang digunakan dalam berkomunikasi. Dimana dalam komunikasi tersebut akan dilihat seberapa jauh interaksi yang dilakukan melalui smartphone dapat membuat penggunanya menjadi ketergantungan. Baik melalui komunikasi telepon, sms, massanger (BBM, line, kakao) dan internet.

Fitur-fitur pada smartphone antara lain:

1. SistemOperasi

Ciri utama sebuah smartphone adalah memiliki sistem operasi di dalamnya yang memungkinkan kita menjalankan berbagai aplikasi, misalnya Windows Mobile, Android, Symbian, ataupun Sistem Operasi Blackberry.

2. Software / Aplikasi

Sebuah smartphone selalu dilengkapi berbagai aplikasi/software yang tentunya ditujukan untuk meningkatkan produktivitas dan mendukung kegiatan sehari-hari. Misalnya Doc To Go, sebuah aplikasi untuk membuat dan mengedit dokumen word di smartphone.

3. Mengakses Internet / Web

Kemampuan lain yang dimiliki oleh sebuah smartphone adalah bisa digunakan untuk mengakses web / internet.

Konten yang disajikan di broswer nya, sudah hampir mendekati seperti layaknya kita mengakses web lewat komputer. Opera Mobile, SkyFire Mobile, IE Mobile adalah contoh beberapa browser di sebuah smartphone.

4. Keyboard QWERTY

Ciri khas lainnya dari smarphone adalah QWERTY keyboard.

Ini tentunya untuk mempermudah penggunanya mengetik dokumen atau mengirim pesan. Tampilan QWERTY keyboard bisa dalam bentuk fisik (hardware) misalnya seperti pada Blackberry, juga bisa tampil dalam bentuk Keyboard virtual seperti pada iPhone.

5. Messaging

Kemampuan mengolah pesan pada smartphone tidak hanya terbatas pada mengirim sms, tapi juga telah dilengkapi kemampuan mengirim email. Sehingga dengan mudah kita bisa mengakses pesan yang sama baik lewat smartphone maupun komputer kita.

6. Multi tasking

Kemampuan sebuah smartphone dalam mengakses banyak fitur di satu waktu, sangat bergantung dengan Sistem Operasi yang tertanam didalamnya. fitur multi tasking dimana user dapat mengakses berbagai aplikasi dalam satu waktu.

7. Touch screen

Dengan teknologi ini pengguna tidak memerlukan keypad untuk menulis pesan atau menelpon. Cukup hanya dengan menyentuhkan jari pada layar smartphone. Teknologi ini sangat identik dengan Iphone sebagai smartphone touch screen paling populer di dunia.

8. Wi-fi

Teknologi dengan nama panjang wireless fidelity ini memungkinkan pengguna smartphone untuk terhubung dengan internet tanpa menggunakan kabel penghubung internet. Dengan fitur ini pengguna smartphone dapat browsing internet dengan leluasa.

9. GPRS

Dengan menggunakan teknologi GPRS (General Packet Radio Service), pengguna dapat dengan optimal menggunakan fasilitas google dan peta sebagai penunjuk jalan pada saat bepergian. (Sari,
2005)

2.8 PHP (*Hypertext Preprocessor*)

Menurut Arief (2011) "PHP (Perl Hypertext Preprocessor) adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis". Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis.

Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.

2. PHP dapat berjalan pada web server yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat open source.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembanganya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 operating sistem, yaitu: Linux, unux, dan windows, dan juga dapat dijalankan secara runtime pada suatu console.

2.9 HTML

Menurut Arief (2011) "HTML (HyperText Markup Language) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman web".

2.10 MySQL

Menurut Arief (2011) MySQL (My Structure Query Languange) adalah "salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya". Mysql bersifat open sourcedan menggunakan SQL (Structured Query Languange). MySQL biasa dijalankan diberbagai platform misalnya windows Linux, dan lain sebagainya.

MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis di bawah lisensi GNU General Public Licence (GPL). Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau open source sehingga kita bisa menggunakannya secara gratis.

Adapun kelebihan-kelebihan dari MySQL yaitu:

1. Source MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.

3. Pengaksesan database dapat dilakukan dengan mudah.
4. MySQL merupakan program yang multithreaded, sehingga dapat dipasang pada server yang memiliki multiCPU.
5. Didukung program program umum seperti C, C++, Java, Perl, PHP, Python, dsb.
6. Bekerja pada berbagai platform. (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
8. Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
9. Mendukung ODBC untuk sistem operasi Windows.
10. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

2.11 Internet

Menurut Sibero (2011) "Internet (Interconneted Network) adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet dapat juga dapat disebut jaringan alam suatu jaringan yang luas". Seperti halnya jarigan komputer lokal maupun jaringan komputer area, internet juga menggunakan protokol komunikasi yang sama yaitu TCP/IP (Tranmission Control Protol / Internet Protocol)".

2.12 Basis Data (*Database*)

Menurut Indrajani (2011) beberapa pengertian basis data, yaitu :

a. Sebuah kumpulan data yang berhubungan secara logis dan merupakan penjelasan dari data tersebut yang dirancang dengan tujuan untuk menemukan data yang dibutuhkan oleh suatu perusahaan atau organisasi. Basis data juga dapat dikatakan sebagai kumpulan data yang selaing terintegrasi karena basis data dirancang untuk dapat digunakan oleh banyak pemakai, memegang data operasional dan juga penjelasan mengenai data tersebut, dan menghindari duplikasi data.

b. Sebuah kumpulan elemen data yang terintegrasi dan berhubungan secara logika. Basis data menggabungkan berbagai catatan yang sebelumnya disimpan dalam *file* terpisah ke dalam suatu elemen data.

2.12.1 Perancangan Basis Data

Menurut Indrajani (2011) Perancangan basis data terdiri dari tiga fase, yaitu :

1. *Conceptual Database Design* merupakan suatu proses pembentukan model yang dibangun dengan menggunakan informasi dalam
2. spesifikasi kebutuhan *user* dan merupakan sumber informasi untuk fase desain logikal.
3. *Logical Database Design* merupakan proses pembentukan model dengan memperbaiki dan memetakan model data konseptual yang telah dibuat sebelumnya.
4. *Physical Database Design* merupakan proses yang menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder, gambaran struktur penyimpanan, dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Desain fisikal juga bisa dikatakan sebagai cara pembuatan menuju suatu DBMS tertentu.

2.13 Database Management System (DBMS)

DBMS adalah sebuah piranti lunak yang mengatur penyimpanan dan akses data di dalam database. Tentunya hal ini di atur secara otomotis – berdasarkan sistem informasi management. Kata management meliputi beberapa fungsi yang meliputi keamanan, akses secara bersama – sama, dan perbaikan. Pengguna dan program meng-akses dan menyimpan data dengan berinteraksi melalui DBMS.

Kebanyakan dari DBMS mempunyai beberapa cara untuk menyimpan data pada alat penyimpanannya, seperti bagaimana data di atur dalam disk, bagaimana cara meng-aksesnya, dan seterusnya dan semuanya berinteraksi dengan level internal daridatabase.

DBMS bertanggungjawab dalam membagi aplikasi atau pengguna (sesuai dengan hak akses) dari keruwetan level internal dan menampilkan secara mudah, dalam

tampilan logika dari sebuah data. Tujuan dari semua database adalah memberikan dukungan data fisik yang berdiri sendiri (physical data independent). Istilah lain adalah penampilan luar (eksternal) yaitu bagaimana menampilkan data kepada masing – masing pengguna.

Database juga mempunyai sebuah metadata (data dictionary) yang berfungsi untuk menyimpan informasi tentang database dan objek – objek yang ada di database.

2.14 Referensi

- **Linggar Arief P, 1211050183, Judul skripsi "**Perancangan Sistem Informasi Penjadwalan Sidang Skripsi dan Praktek Kerja dan Pengabdian Masyarakat (PKPM) Program Studi Sistem Informasi (SI) Berbasis Web pada IBI Darmajaya Bandar Lampung.
- Galoh Randicha, Wahyul A S dan Adian Fatchur R, 2012. *Sistem Penjadwalan Sidang Tugas Akhir Berbasis Web Dengan Pesan Pengingat Melalui Sms Dan Aplikasi Pada Perangkat Android Di Jurusan Teknik Elektro Universitas Diponegoro,* Jurnal Sistem Informasi. Universitas Diponegoro. Semarang.