BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Perkembagan dunia Informasi yang sangat cepat, menuntut para semua kalangan mulai dari kalangan bawah atau kalangan atas untuk selalu memperbaharui informasi mereka setiap saat, dimanapun dan kapanpun, oleh karna itu pada era sekarang ini sangat dibutuhkan suatu media yang dapat menyampaikan suatu informasi yang sangat cepat dan tepat serta efisien, sms gateway merupakan media yang tepat dalam hal ini karena sms gateway hanya membutuhka suatu id yang berupa nomor telepon, maka informasi tersebut dapat terkirim dengan cepat.

Ada beberapa penelitian yang pernah dilakukan megenai tentang sms gateway itu sendiri, berikut adalah hasil atau kesimpulan dari penelitian tentang sms gateway.

Rio Andrian (*2014 :23)* Mengungkapkan bahwasanya, *Short message service* (SMS) adalah teknologi yang sangat banyak diminati dan digunakan oleh banyak kalangan masyarakat. Selain karena unggul dari segi kepraktisan dan kemudahan dalam penggunaannya, teknologi ini juga hadir dengan tarif yang relatif lebih murah untuk fasilitas pengiriman data pesan atau transfer informasi dalam kapasitas kecil dibandingkan dengan layanan, sistem penyebaran informasi baru berbasis SMS *Gateway* sangat dibutuhkan guna memperbarui sistem penyebaran informasi yang telah ada saat ini di sekolah agar penyampaian

informasi lebih cepat dan efisien. Informasi dapat diterima langsung ke ponsel calon siswa melalui pesan singkat**.**

Menurut anjar priyadna (2013:24), "SMS *Gateway* adalah jenis layanan dua arah artinya selain dapat menerima pesan dari luar juga dapat mengirim balasan secara otomatis ke nomor tujuan contohnya Sms Quis, Sms Polling dll".

Karena sifatnya yang dua arah, maka jenis sms ini sangat cocok digunakan sebagai SMS *Center* sebuah organisasi atau perusahaan dalam rangka meningkatkan kualitas komunikasi antara anggota komunitas organisasi atau pegawai di dalam perusahaan (*Muhammad Taufiq Muslih, 2013).*

2.2 Teori Pendukung

A. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan didalam pendefinisian sistem, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya.

Sistem merupakan bagian terpenting dalam perkembangan sehingga para ahli mengalihkan perhatian kepada pembelajaran mengenai sistem.

1. Pengertian Sistem

Menurut Jerry Fitzgrald, et, al dalam Puspitawati dan Anggadini (2012:1) "Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu".

Menurut Puspitawati dan Anggadini (2012:2) "Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu".

Menurut Azhar Susanto (2013:3) “Sistem adalah suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain”.

Sedangkan menurut Norman L.Enger dalam Sutabri (2012:7) “Sistem adalah sesuatu yang terdiri atas kegiatan-kegiatan yang berhubugan guna mencapai tujuan-tujuan perusahaan seperti pengendalian inventaris atau penjadwalan produksi”.

2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus.

Sistem mempunyai karakteristik atau sifat-sifat tertentu menurut Sutabri (2012:13), yaitu:

a. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

b. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan Luar Sistem (*Environtment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

d. Penghubung Sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan sub sistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain.

e. Masukan Sistem (*Input*)

Energi yang dimasukan kedalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

g. Pengolahan Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

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

3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya sebagai berikut:

a. Sistem Abstrak (*Abstract Sistem*) dan Sistem Fisik (*Physical Sistem*)

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu suatu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

b. Sistem Alamiah (*Nature Sistem*) dan Sistem Buatan (*Human Made Sistem*)

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem peputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machine system*. Sistem informasi bebasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem Tertentu (*Deterministic Sistem*) dan Sistem Tak Tentu (*Probabilistic Sistem*)

   Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-

program komputer yang dijalankan. Sedangkan sistem yang bersifat *probabilistic* adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

d. Sistem Tertutup (*Closed Sistem*) dan Sistem Terbuka (*Open Sistem*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

4. Pengertian Informasi

Menurut Mardi (2013:5) mengemukakan "Informasi adalah hasil proses atau hasil pengolahan data melalui gabungan, analisis, penyimpulan dan pengolahan sistem informasi komputerisasi".

Informasi Menurut Gordan B. Davis dalam Sutabri (2012:1) adalah " Informasi adalah data yang telah diproses kedalam bentuk yang mempunyai arti bagi si penerima, dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau keputusan mendatang".

Menurut Sutabri (2012:29) adalah "Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan".

Kualitas Informasi tergantung dari 3 (tiga) hal yaitu:

1. Akurat (*Accurate*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas dan mencerminkan maksudnya. Informasi harus akurat karena biasanya dari sumber informasi sampai penerima informasi ada kemungkinan terjadi gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut.

2. Tepat Waktu (*Timelines*)

   Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi. Dewasa ini, mahalnya informasi disebabkan karena harus cepatnya informasi tersebut dikirim atau didapat sehingga diperlukan teknologi mutakir untuk mendapat, mengolah, dan mengirimkannya.

3. Relevan (*Relevance)*

   Informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevansi informasi untuk orang satu dengan yang lain berbeda, misalnya informasi sebab musabah kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan apabila ditujukan kepada ahli teknik perusahaan. Sebaliknya, informasi mengenai harga pokok poduksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan relevan untuk seorang akuntan perusahaan.

4. Pengertian Sistem Informasi

Menurut Sutabri (2012:38) "Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu".

Menurut Azhar Sutanto dalam Puspitawati dan Anggadini (2012:14) "Sistem Informasi merupakan komponen-komponen dari subsistem yang saling berhubungan dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi".

Sedangkan menurut John Burch dan Gary Grudnitski dalam Puspitawati dan Anggadini (2012:20) "mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input blok*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*controls blok*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

2.2.Teori Pendukung

A. Unifiled Modeling Diagram (*UML*)

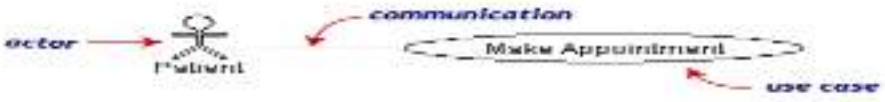
Menurut Widodo (2012:10) Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi,

diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. **Diagram Use Case (*Use Case Diagram)***

   Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu *apa yang dilakukan* bukan *bagaimana melakukannya*. Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. untuk lebih memperjelas lihat gambaran suatu peristiwa untuk sebuah klinik kesehatan di bawah ini :

   "Pasien menghubungi klinik untuk membuat janji (*appointment*) dalam pemeriksaan tahunan. *Receptionist* mendapatkan waktu yang luang pada buku jadwal danmemasukkan janji tersebut ke dalam waktu luang itu".



Gambar II.1
Contoh kegiatan pasien yang membuat janji

Diagram Use Case berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (*requirements*)

   Use Case baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan *design* menjadi lebih jelas.

2. Komunikasi dengan klien

   Penggunaan notasi dan simbol dalam diagram *Use Case* membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.

3. Membuat test dari kasus-kasus secara umum

   Kumpulan dari kejadian-kejadian untuk *Use Case* bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

2. Diagram Kelas (*Class Diagram)*

Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.

Diagram Class mempunyai 3 macam *relationalships* (hubungan), sebagai berikut:

1. ***Association***

   Suatu hubungan antara bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.

2. ***Aggregation***

   Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh : OrderDetail merupakan kumpulan dari Order.

3. ***Generalization***

   Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *superClass* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *superClass*.

   Package dan *Object* Untuk mengatur pengorganisasian diagram Class yang *kompleks*, dapat dilakukan pengelompokan kelas-kelas berupa *packag*e (paketpaket). *Package* adalah kumpulan elemen-elemen logika UML.

3. Diagram Interaksi dan *Sequence (Sequance Diagram)*

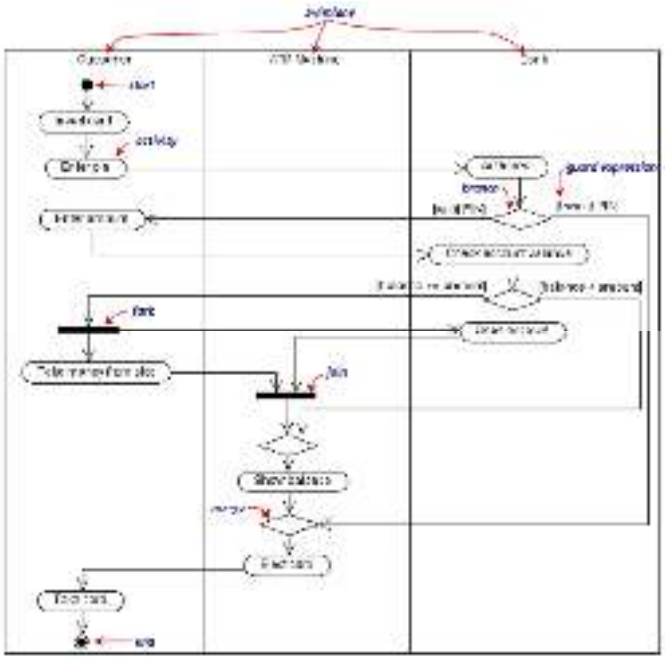
Diagram Class dan diagram Object merupakan suatu gambaran *model statis*. Namun ada juga yang bersifat *dinamis*, seperti Diagram Interaction. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

4. **Diagram *StateChart (StateChart Diagram)***

Behaviors dan *state* dimiliki oleh obyek. Keadaan dari suatu obyek bergantung pada kegiatan dan keadaan yang berlaku pada saat itu. Diagram StateChart menunjukan kemungkinan dari keadaan obyek dan proses yang menyebabkan perubahan pada keadaannya.

5. Diagram Activititas *(Activity Diagram)*

Pada dasarnya diagram Activity sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada *obyek yang dalam suatu proses* (atau proses menjadi suatu obyek), diagram Activity berfokus pada *aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal*. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Sebagai contoh, perhatikan proses yang terjadi. "Pengambilan uang dari bank melalui ATM." Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait, yaitu Customer, ATM, and Bank. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Lihat gambar di bawah ini, agar lebih jelas.



Gambar II.2
Contoh Diagram Activity 'Pengambilan Uang melalui ATM'

Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif).

Label *Guard Expression* (ada di dalam [ ]) yang menerangkan output (keluaran) dari percabangan. percabangan akan menghasilkan bentuk menyerupai bentuk intan. *transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut **Fork.** *Fork* beserta *join* (gabungan dari hasil output *fork*) dalam diagram berbentuk *solid bar* (batang penuh).

6. **Diagram komponen *(Component Diagram)***

   Bersifat statis, Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

7. **Diagram deployment *(deployment diagram)***

   Bersifat statis, Diagram inimemperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

8. **Diagram komunikasi *(Communication Diagram)***

   Bersifat dinamis, Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

9. **Diagram paket *(Package Diagram)***

   Bersifat statis, Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen

Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML, seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

a. Building blocks

Tiga macam yang terdapat dalam building block adalah :

1. **Benda *Things***

   Adalah abstraksi yang pertama dalam sebuah model

2. **Hubungan*/Relationships***

   Sebagai alat komunikasi dari benda-benda

3. **Bagan*/Diagrams***

   Sebagai kumpulan / group dari benda-benda/things

b. Benda/*Things*

Adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemenelemen

lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda / thing adalah sebagai berikut :

1. ***Classes,*** yang diuraikan sebagai sekelompok dari *object* yang mempunyai *atribute*, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*. Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, *atribute*, dan metoda pengoperasiannya.
2. ***Interfaces,*** merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. *Interface* / antar-muka mendefinisikan sebuah set / kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen.
3. ***Collaboration,*** yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas / elemen-elemen yang bekerja secara bersama-sama. *Collaborations* mempunyai struktur dan dimensi. Pemberian sebuah kelas memungkinkan berpartisipasi didalam beberapa *collaborations* dan digambarkan dengan sebuah '*elips*' dengan garis terpotong-potong.
4. ***Use cases,*** adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. '*use case*' digunakan untuk membentuk tingkah-laku benda / *things* dalam sebuah model serta di realisasikan oleh sebuah

collaboration. Umumnya '*use case*' digambarkan dengan sebuah '*elips*' dengan garis yang solid, biasanya mengandung nama.

5. ***Nodes,*** merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalaha sebuah komputer, umumnya mempunyai sedikitnya *memory* dan *processor*. Sekelompok komponen mungkin terletak pada sebuah node dan juga mungkin akan berpindah dari node satu ke node lainnya. Umumnya node ini digambarkan seperti kubus serta hanya mengandung namanya.

c. Hubungan / *Relationship*

Ada 4 macam hubungan didalam penggunaan UML, yaitu;

1. ***Dependency,*** adalah hubungan semantik antara dua benda/*things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus-putus.
2. ***Association*****,** hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *assosiation* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.
3. ***Generalizations*****,** adalah menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent* / induk . Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan

tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah.

4. ***Realizations*****,** merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan *realization*.

B. Entity Relationship Diagram (*ERD*)

Menurut Sutanta (2012:91) "*Entity Relationship Diagram* (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek." *Entity Relationship Diagram* (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram* (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antardata didalamnya.

Komponen Entity Relationship Diagram menurut Sutanta (2011:91) adalah sebagai berikut :

a. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut :

1) Entitas dinyatakan dengan simbol persegi panjang.
2) Nama entitas dituliskan didalam simbol persegi panjang.
3) Nama entitas berupa kata benda, tunggal
4) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

Ada dua macam entitas yaitu:

1. *Entity/Entity Strong* (Entitas Kuat)

   Entitas kuat merupakan entitas yang tidak memiliki ketergantungan terhadap entitas lainnya.

2. *Entity weak* (Entitas lemah)

   Sedangkan entitas lemah merupakan entitas yang keberadaannya tergantung terhadap keberadaaan entitas lain dalam suatu relasi.

b. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

1) Atribut digambarkan dengan simbol ellips.
2) Nama atribut dituliskan didalam simbol elips

3) Nama atribut merupakan kata benda, tunggal.
4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

c. Relasi

Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :

1) Relasi dinyatakan dengan simbol belah ketupat
2) Nama relasi dituliskan didalam simbol belah ketupat
3) Nama relasi berupa kata kerja aktif.
4) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas

Di dalam relationship ada 3 bentuk hubungan yang biasa disebut dengan kardinalitas. Kardinalitas Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

Macam-macam kardinalitas adalah:

1. **Satu ke satu** ***(One to one)***

   Hubungan relasi satu ke satu, yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2. **Satu ke banyak** ***(One to many)***

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. **Banyak ke banyak** ***(Many to many)***

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

Komponen penyusun ERD adalah sebagai berikut :

Notasi	Keterangan
ENTITAS	**Entitas**, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
RELASI	**Relasi**, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
ATRIBUT	**Atribut**, berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Gambar II.3
Komponen Penyusun ERD

Berikut adalah metode/tahap untuk membuat ERD

a. Menentukan Entitas
b. Menentukan Relasi
c. Menggambar ERD sementara
d. Mengisi Kardinalitas
e. Menentukan Kunci Utama
f. Menggambar ERD berdasar Key
g. Menentukan Atribut
h. Memetakan Atribut
i. Menggambar ERD dengan Atribut

C. SMS (Short Message Services)

Short Message Services (SMS) atau layanan pesan singkat merupakan sebuah revolusi di media penyebaran informasi, dimana layanan yang digunakan tidak berbasis suara tetapi berbasis teks singkat.

1. SMS Gateway

Pada prinsipnya, SMS *Gateway* adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistnbusikan pesan-pesan yang di-*generate* lewat sistem informasi melalui media SMS yang di-*handle* oleh jaringan seluler . SMS *Gateway* ini memanfaatkan modem untuk server pengiriman SMS. SMS memanfaatkan jaringan operator seluler untuk penginman sms, *service* gammu sebagai software sms *gateway*, dan *database* mysql yang di integrasikan dengan *database (Rhyca Putri Ardy :2013).*

2. Gammu

Menurut Michail Cihar (2017:1), Gammu adalah sebuah library atau sebuah perintah garis pengguna untuk handphone, di keluarkan dalam GNUGPL, versi 2.0, Gammu telah di resmikan oleh Marcin Wiacek dan tim, sebenarnya Gammu code sendiri adalah hasil dari pengembangan code yang ada di Gnokii dan kemudian My Gnokii projek, Gammu yang diatas versi 0.58 biasa disebut juga sebagai MyGnokii 2.

Dalam penggunaanya gammu memperlukan sebuah file konfigurasi untuk dapat membangnun komunikasi dengan modem / handphone yang digunakan dan file yang dikonfigurasikan adalah gammu/config/gammurc, untuk peng-

konfigurasian koneksi handphone dapat dilihat di https:wammu.eu/phones/ dan gammu/config/gammurc menggunakan ini file syntax.

Dalam konfigurasi file untuk gammu terdapat beberapa sesi – [gammu], [gammu1], [gammuN], setiap sesi konfigurasi koneksi, dan [gammu] mempunyai nilai index [0] atau awal dari sesi / koneksi pertama (default) yang mana nantinya akan digunakan ketika service dari gammu dijalankan.

3. Parameter koneksi alat (modem / handphone)

Protokol yang mana biasanya digunakan untuk membangun komunikasi dengan modem / handphone yang digunakan:

a. koneksi (connection)

Dengan menggunakan kabel hp nokia

Koneksi	Tipe
dlr3	Kabel DLR-3
dku2	Kabel DKU-2
dku5	Kabel DKU-5
Mbus	Koneksi mbus
Fbus	Koneksi fbus

Tabel II.1
Kabel Handphone Nokia

At	Berdasarkan koneksi(biasanya tergantung kecepatan koneksi dari modem / hp)
Irdaphonet	Koneksi untuk handphone nokia

Tabel II.2
Koneksi Handphone Nokia

Contoh untuk pengkonfigurasian koneksi dalam file gammurc :

connection = a19200 nokia

1. **Alat (Device)**

   Dalam hal pengkonfigurasian alat / device dalam file gammurc tergantung pada serial port yang digunakan, dan kemudian masuk serial port dalam file gammurc, berikut adalah contoh pengkonfigurasian :

   a. Dalam sistem operasi linux :

      Device = /dev/ttyACM0

   b. Dalam sistem operasi windows:

      Device = COM5

2. **Model**

   Parameter model jarang digunakan dalam pengkonfigurasian file gammurc, parameter model digunakan ketika Gammu tidak mengetahui tipe dari modem/ handphone yang digunakan, atau ketika ada features yang tidak dapat digunakan.

   berikut contoh settingan parameter model dalam file gammurc :

   Model = 6110

Dengan demikian berikut adalah contoh dalam file gammurc yang dapat digunakan utnuk tipe yang sesuai dengan koneksi tersebut:

```
device = COM5
connection = at19200
model = 6110
```

Gambar II.4
Konfigurasi File gammurc

D. Php (Hypertext Preprocessor)

Menurut Irwin Nughroho (2011:11), PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. PHP digunakan sebagai bahasa *script server-side* dalam pengembangan Web yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan Web dapat dibuat dinamis sehingga *maintenance* situs Web menjadi lebih mudah dan efisien. PHP ditulis menggunakan bahasa C.

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih daripada kemampuan CGI. PHP tidak terbatas pada hasil keluaran HTML *(HyperText Markup Language)*. PHP juga memiliki kemampuan untuk mengolah gambar, file PDF, dan movie flash. PHP juga dapat menghasilkan teks seperti HTML dan file XML lainnya. Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungannya terhadap banyak database, salah satunya adalah MySQL.

a. Sintaks PHP

Sintaks program/*script* PHP ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok *script* PHP:

1) <?php ... ?>

2) <script language = “PHP”> ... </script>

3) <? ... ?>

Cara 1 dan 2 merupakan cara yang paling umum digunakan sekalipun cara tampak lebih praktis karena cara 3 tidak selalu diaktifkan pada konfigurasi file php.ini. Sedangkan cara 4 dimungkinkan sebagai kemudahan bagi yang sudah terbiasa dengan ASP *(Active Server Pages).* Namun, bila ini tidak dikenal, maka harus dilakukan pengaktifan pada file konfigurasi php.ini.

b. **Menampilkan *String***

Untuk menampilkan *string* dalam PHP disediakan fungsi seperti ditunjukkan pada tabel berikut:

Fungsi Sintaks	Sintaks
Echo	echo (string arg1 [,string argn])
Print	print (string format [, mixed args])
Printf	printf (string format [, mixed args])

Tabel II.3
Fungsi Menampilkan String dalam PHP

c. Struktur Kontrol

1) Statement *if*

Statement *if* digunakan untuk mengeksekusi sebuah *blok pernyataan* jika memenuhi kondisi tertentu.

Sintaksnya:

```
if (kondisi)
   blok pernyataan
```

Gambar II.5
Statement If

Jika kondisi bernilai *true* (benar), *blok pernyataan* akan dikerjakan. Apabila pernyataan yang dikerjakan lebih dari satu, maka harus diletakkan dalam tanda {}.

2) Statement *if ... else ...*

Perintah *if ... else ...* pada prinsipnya mirip dengan perintah *if*, tetapi ada kalanya anda menginginkan dua percabangan, yakni jika suatu kondisi terpenuhi, maka lakukan *blok pernyataan1*. Jika tidak terpenuhi, lakukan *blok pernyataan2*. Untuk kebutuhan tersebut, gunakan statement *if ... else*.

Sintaksnya:

```
if (kondisi)
{
   blok pernyataan1;
}
else
{
   blok pernyataan2;
}
```

Gambar II.6
Statement If Else

Jika kondisi bernilai *true* (benar), maka *blok pernyataan1* akan dikerjakan. Jika bernilai *false* (salah), maka *blok pernyataan2*-lah yang akan dikerjakan.

3) Statement *if ... elseif ... else ...*

Statement *if ... elseif ... else ...* digunakan untuk masalah yang membutuhkan lebih dari dua percabangan. Statement *if ...elseif ... else ...* sering disebut *nested if* (*if* bersarang).

Sintaksnya:

```
if (kondisi1)
{
   blok pernyataan1;
}
elseif (kondisi2)
{
   blok pernyataan2;
}
...
else
{
   blok pernyataanN;
}
```

Gambar II.7
Statement If else If

Jika *kondisi1* bernilai *true*, maka *blok pernyataan1* akan dikerjakan. Jika *false*, maka diuji *kondisi2*. Jika *kondisi2* bernilai *false*, maka diuji kondisi berikutnya. Namun, jika tidak ada kondisi yang terpenuhi, maka akan dikerjakan *blok pernyataan* ke-*N*.

4) Statement *while*

Statement *while* adalah statement yang digunakan untuk melakukan perulangan mengevaluasi blok pernyataan selama kondisi *true* (benar), dan akan berhenti apabila kondisi bernilai *false* (salah).

Sintaksnya:

```
while (kondisi)
{
   blok pernyataan;
}
```

Gambar II.8
Statement While

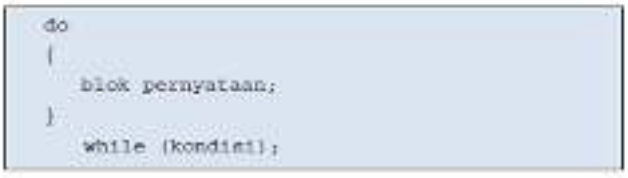
a. *kondisi* adalah pernyataan *boolean*.

b. *blok pernyataan* adalah daftar statement yang akan diulangselama kondisi terpenuhi.

5) Statement *do ... while*

Pada prinsipnya statement *do ... while* sama dengan cara kerja *while*, hanya saja pada *do ... while* blok pernyataan pasti dikerjakan sekali dan kemudian dilakukan pengujian kondisi. Jika kondisi masih terpenuhi *(true),* maka blok pernyataan dikerjakan lagi. Namun, jika kondisi tidak terpenuhi *(false)* lagi, maka perulangan berhenti.

Sintaksnya:



Gambar II.9
Statement do

a. kondisi adalah bernilai *boolean.*

b. *blok* pernyataan adalah daftar statement yang diulang selama kondisi dipenuhi (benar). Jika kondisi tidak terpenuhi (salah), maka anda bisa keluar dari perulangan dan mengerjakan statement setelah *while*.

6) Statement *for*

Statement *for* adalah statement yang digunakan untuk mengulang *blok pernyataan* dalam jumlah yang ditentukan berdasarkan

inisialisasi awal, akhir/kondisi, dan nilai penambahan atau pengurangan yang ditentukan.

Sintaksnya:

```
for (inisialisasi; kondisi; increment)
{
   blok pernyataan;
}
```

Gambar II.10
Statement For

Statement *for* bekerja sebagai berikut:

a. *inisialisasi* sebagai nilai awal.
b. kondisi diuji; jika bernilai *true* (benar), maka perulangan dilanjutkan dengan mengerjakan *blok pernyataan,* sedangkan jika bernilai *false* (salah), maka perulangan berhenti dan *blok pernyataan* dilompati.
c. Jika *blok pernyataan* hanya terdiri dari satu baris, maka tanda {} dapat ditiadakan.
d. *increment* merupakan nilai penambahan atau pengurangan untuk mengulangi pengerjaan *blok pernyataan* setelah penambahan atau pengurangan yang nilai kebenarannya diuji apakah kondisi masih terpenuhi.
e. *string filename* menyatakan nama *file* yang akan digabungkan.

7) Statement *require*

Statement *require()* merupakan konstruksi bagi parser PHP yang digunakan untuk membuka file yang diberi dan membaca nilai variabel serta fungsi yang terdapat didalamnya untuk kemudian

mengeksekusinya. File akan diperlakukan sebagai suatu *script* PHP normal. Apabila file tersebut berisi tag-tag PHP, maka akan dievaluasi terlebih dahulu sebelum mengirimnya ke *browser,* tetapi apabila hanya berisi teks biasa, maka akan dikirim langsung ke *browser*. Statement *require()* tidak dapat dimasukkan ke dalam struktur perulangan karena hanya boleh dipanggil satu kali.

Sintaksnya:

```
require (string filename)
```

Gambar II.11
Statement require

8) Statement *include*

Statement *include()* merupakan konstruksi bagi parser PHP yang digunakan untuk membuka dan membaca nilai variabel dari file yang dinyatakan serta fungsi yang tedapat didalamnya untuk kemudian mengeksekusinya.

File akan diperlakukan sebagai suatu *script* PHP normal. Apabila file tersebut berisi tag-tag PHP, maka akan dievaluasi terlebih dahulu sebelum mengirimnya ke *browser,* tetapi apabila hanya berisi teks biasa, maka akan dikirim langsung ke *browser*. *Statement include()* dapat dimasukkan dala struktur perulangan.

Sintaksnya:

```
include (string filename)
```

Gambar II.12
Statement inlcude

a. *string filename* menyatakan nama *file* yang akan digabungkan.

d. Session

Session dalam PHP dapat dapat dimulai dengan dua cara, yaitu secara otomatis dan menggunakan fungsi *session* pada *script* PHP. Untuk memulai *session* secara otomatis, file php.ini perlu diedit dengan melakukan perubahan pada baris *session.auto.start* = 0 menjadi *session.auto.start = 1*, kemudian simpan perubahan tersebut dan *restart* kembali *web server.* Untuk memulai *session* menggunakan fungsi pada PHP, gunakan fungsi *session_start().*

Penggunaan *session* dengan *session_start()* akan menghasilkan file *session* dengan nama *sess_* diikuti oleh nilai *session_id*. Session dapat juga dibuat dengan menggunakan fungsi *session_register().* Penggunaan *session_register()* PHP memungkinkan penyimpanan variabel dan nilainya dalam file.

Untuk mengakhiri *session* digunakan fungsi *session_destroy().* Sedangkan untuk menghapus semua variabel *session*, digunakan fungsi *session_unset().* Sementara itu, untuk menghapus sebuah variabel dari sebuah *session* dan agar *session* tetap ada, dapat digunakan fungsi *session_unregister(nama variabel).*

2.3. Penelitian Terkait

Banyak penelitian yang sebelumnya dilakukan mengenai sistem informasi yang menggunakan *sms gateway* dan penelitian lain yang berkaitan. Dalam upaya mengembangkan dan menyempurnakan sistem informasi yang menggunakan *sms gateway* ini perlu dilakukan studi pustaka (*literature review*) sebagai salah satu dari penerapan metode penelitian yang akan dilakukan, diantaranya yaitu :

Pada tahun 2016 Imam Hambali dan Mahmud Yunus, melakukan sebuah penelitian di PT. Mega Finance cabang malang, dalam penelitian tersebut dikatakan bahwasanya yang menjadi akar masalah adalah tidak terkontrolnya pekerjaan dilapangan(*field*) oleh atasan secara langsung karena banyaknya karyawan dilapangan (*field*) menyebabkan sering terjadinya *fraud* atau kecurangan yang dilakukan oleh karyawan pada PT. Mega Finance cabang malang, akibat dari banyaknya *fraud* maupun *lapping* pada perusahaan pembiayaan, mengharuskan perusahaan pembiayaan untuk mengatasi masalah penyampaian informasi pemasaran maupun tagihan yang benar pada customernya secara langsung. dalam penelitian tersebut peneliti merumuskan masalah yang terdapat pada PT. Mega Finance cabang malang, berikut adalah rumusan masalah, Membangun SMS Gateway Manajemen Collection Mega Finance Cabang Malang, untuk mendapatkan mengetahui kondisi di PT. Mega Finance cabang malang serta mendapatkan data (masalah) peneliti menggunakan metode wawancara secara langasung kepada karyawan di PT. Mega Finance cabang malang, adapun kesimpulan dalam penelitian tersebut, aplikasi SMS Gateway Mega Financedapat menyampaikan informasi promo maupun tagihan yang tepat dan secara langsung pada cutomer Mega Finance serta dapat membantu dan

memudahkan customer dalam menyampaikan kritik dan saran seputar kreditnya di PT.Mega Finance kantor cabang malang. (Imam Hambali, Mahmud Yunus 2016: 4).

Pada tahun 2014 mochammad Rendy Rikskanto Widodo, M. Roziq Zainuddin, laura saraswati nusantara membuat sebuah penelitian dengan judul "Sistem Informasi Dan Pengolahan Data Kursus Mobil Berbasis Web Dengan Sms Gateway Di Armada Pasuruan" menurut penelitian tersebut dikatakan bahwasanya, pengembangan suatu sistem informasi dilakukan dengan tujuan mengubah proses manual menjadi komputerisasi sehingga meningkatkan kinerja dan mempermudah dalam memanajemen, dalam sistem yang dibuat pada penelitian tersebut terdapat 2 level hak akses yang pertama adalah admin dan kedua adalah *User*/siswa, dalam penelitian tersebut juga dijabarkan tentang hak akses yang dimiliki keduanya, untuk admin dimana pengguna ini mempunyai akses semua proses pengolahan data seperti data siswa, data mobil, data paket, data kursus, data pembanyaran, melakukan pengiriman SMS serta membuat laporan kepada pihak pemilik lembaga. Sedangkan pengguna user/siswa dimana mempunyai akses hanya untuk data khusus user/siswa tersebut seperti data siswa, data kursus, data pembayaran serta data jadwal,adapun kesimpulan pada penelitian tersebut adalah Sistem informasi dan pengolahan data kursus mobil berbasis web ini mudah digunakan dan diakses dimana saja dan kapan saja karena menggunakan system online. Proses pendaftaran yang semakin mudah dengan memanfaatkan halaman user pengguna sekaligus sebagai media informasi mengenai data kursus dari masing masing siswa yang telah terdaftar. Penyampaian informasi tentang jadwal kursus dan penagihan biaya kursus lebih

cepat karena sudah tersedianya sistem reminder dengan layanan SMS *Gateway*. Lebih cepat dan efisien dalam pengelolaan data terkait kegiatan administrasi data kursus. Memudahkan dalam mengelola dan menentukan jadwal kursus masing masing siswa. (mochammad Rendy Rikskanto Widodo, M. Roziq Zainuddin, laura saraswati 2014: 45)