

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Kata akademik berasal dari bahasa Yunani yakni *academos* yang berarti sebuah taman umum (plasa) di sebelah barat laut kota Athena. Nama *Academos* adalah nama seorang pahlawan yang terbunuh pada saat perang legendaris Troya. Pada plasa inilah filosof Socrates berpidato dan membuka arena perdebatan tentang berbagai hal. Tempat ini juga menjadi tempat Plato melakukan dialog dan mengajarkan pikiran-pikiran filosofisnya kepada orang-orang yang datang. Sesudah itu, kata *acadomos* berubah menjadi akademik, yaitu semacam tempat perguruan. Para pengikut perguruan tersebut disebut *academist*, sedangkan perguruan semacam itu disebut *academia*. Berdasarkan hal ini, inti dari pengertian akademik adalah keadaan orang-orang bisa menyampaikan dan menerima gagasan, pemikiran, ilmu pengetahuan, dan sekaligus dapat mengujinya secara jujur, terbuka, dan leluasa.

Dari pengertian akademik, pengertian sistem, dan pengertian informasi yang telah dibahas maka penulis mengambil kesimpulan mengenai pengertian sistem informasi akademik sebagai berikut :

“Sistem informasi akademik adalah sekumpulan elemen-elemen baik fisik maupun non fisik dan prosedur yang saling berhubungan satu sama lain menjadi satu kesatuan dan bekerja sama untuk mengolah data akademik di sebuah lembaga pendidikan menjadi informasi yang berguna bagi pemakainya.”

Secara umum data-data yang diolah oleh sistem informasi akademik pada suatu sekolah meliputi data pendaftaran siswa baru, data daftar ulang siswa, dan data-data lain yang berhubungan dengan proses ke akademikan.

Berdasarkan penelitian sebelumnya oleh Dengen dan Dyna Marisa Kh (2009) yang diambil dari jurnal sistem informasi akademik berbasis *web* SMP Negeri 4 Samarinda mengemukakan bahwa sistem informasi akademik berbasis web pada SMP Negeri 4 Samarinda merupakan suatu sistem yang memberikan informasi laporan keaktifan siswa secara *online* yang berupa laporan nilai serta laporan absensi siswa yang bersangkutan dengan berbasiskan *web*, sehingga membantu kecepatan dan kualitas dalam penyampaian informasi. Pada hasil penelitian ini telah dikembangkan sebuah Sistem Informasi Akademik Berbasis Web dengan studi kasus pada SMP Negeri 4 Samarinda. Dimana dalam membangun sistem ini digunakan alat bantu pengembangan sistem yaitu Data Flow Diagram (DFD), Context Diagram, Entity Relationship Diagram (ERD) dan Flowchart serta dengan menggunakan bahasa pemrograman PHP dan HTML dan MySQL sebagai databasenya.

Selain itu berdasarkan penelitian oleh Prakoso dan Meliana Christianti (2008) yang diambil dari jurnal Analisis dan Perancangan Sistem Informasi Akademik dengan Studi Kasus pada Sekolah Menengah Atas Terpadu (SMAT) Krida Nusantara mengemukakan bahwa *Sekolah Menengah Atas Terpadu Krida Nusantara is a boarding school located in Cibiru Bandung. The boarding condition and situation made it hard for parents to look over student's grade. For that, SMAT Krida Nusantara needed a system to aid parents in monitoring their children's gradings. Apart from providing information about grades, this system will also provide news and forums facility, to build communication between school and parents. This academic application has functionaliti es to record student grade, subject data, basic competency and indicator, provide informations through News, give 2 way interaction media, storing students data, and calculating average score per class/student.*

A. Konsep Dasar Sistem

Dalam sebuah sistem terdapat dua pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan pada komponen (elemen). Pendekatan sistem yang menekankan pada prosedur yaitu :

”Sistem adalah suatu jaringan dari kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk melakukan suatu kegiatan tertentu. (Jogiyanto, 2005a:1)

Pendekatan sistem merupakan jaringan kerja dari prosedur yang lebih menekankan urutan-urutan operasi.

”Suatu Prosedur adalah suatu urutan-urutan yang tepat dan tahapan-tahapan instruksi yang menerangkan apa (what) yang harus dikerjakan, siapa (who) yang mengerjakan, kapan (when) dikerjakan dan bagaimana (how) mengerjakannya”. (Jogiyanto, 2005b:2)

Sedangkan pendekatan sistem yang menekankan pada elemen, yaitu : ” Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. (Jogiyanto, 2005c:2).

Dapat disimpulkan bahwa sistem adalah himpunan atau grup elemen atau komponen yang berhubungan atau saling bergantung satu sama lain untuk mencapai tujuan tertentu.

Sebuah sistem terdiri dari bagian-bagian saling berkaitan yang beroperasi bersama untuk mencapai sasaran dan maksud. Berarti sebuah sistem bukanlah seperangkat unsur yang tersusun secara tak teratur, tetapi terdiri dari unsur-unsur yang dapat dikenal sebagai saling melengkapi karena mempunyai satu maksud, tujuan atau sasaran.

Menurut Jogiyanto (2005:2) mendefinisikan secara general bahwa “Sistem adalah kumpulan dari elemen–elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu: komponen sistem (*components*), batas sistem (*boundary*), lingkungan luar sistem

(*environment*), penghubung sistem (*interface*), masukan sistem (*input*), keluaran sistem (*output*), pengolahan sistem (*process*), sasaran sistem (*objective*) atau tujuan sistem (*goal*).

B. Konsep Dasar Program

Pada umumnya program adalah sederetan instruksi atau *statement* yang tentunya dalam bahasa yang dimengerti oleh komputer. Instruksi tersebut berfungsi untuk mengatur apa saja yang akan dilakukan oleh komputer agar mendapatkan suatu hasil keluaran yang maksimal. Menurut Kurniawan (2005:1) menyebutkan bahwa “sebuah program pada dasarnya adalah kumpulan instruksi yang menjelaskan kepada komputer untuk melakukan sesuatu”. Instruksi-instruksi yang digunakan disesuaikan dengan jenis bahasa yang digunakan (*reserved word*) yang disediakan.

Kata pemrograman dapat diartikan sebagai cara membuat program, dalam konteks ini berarti membuat program komputer. Dapat juga dikatakan bahwa pemrograman merupakan suatu kumpulan urutan perintah pada komputer untuk mengerjakan suatu bahasa tersendiri yang dapat dibaca atau dimengerti oleh komputer.

Web merupakan kumpulan halaman – halaman untuk menginformasikan sesuatu dalam internet yang dapat berisi teks, gambar, animasi, suara, video dan lainnya. Dalam membangun sebuah web dikenal dengan pemrograman web dikelompokkan menjadi 2 kelompok, yaitu :

1. *Client Side* : Informasi yang disampaikan akan langsung dieksekusi browser pada client. Contoh: HTML, Javaskrip dan lainnya.

2. *Server Side* : Informasi yang dikirim akan dieksekusi di web server sebelum disampaikan ke browser pada client. Contoh: PHP, ASP, JSP dan lainnya.

Aplikasi web diperlihatkan pada gambar. Klien berinteraksi dengan web server. Secara internal, web server berkomunikasi dengan middleware, dan middleware inilah yang berhubungan dengan basis data (database).

Middleware pada gambar tersebut merupakan perangkat lunak yang bekerja sama dengan web server dan berfungsi menerjemahkan kode – kode tertentu, menjalankan kode – kode tersebut dan memungkinkan berinteraksi dengan basis data. PHP, ASP, dan Perl adalah beberapa contoh middleware. Browser atau web browser adalah perangkat lunak di sisi klien yang digunakan untuk mengakses informasi web. Biasanya yang diterima dari web server dalam bentuk HTML. Contohnya: Internet Explorer, Netscape, dan Mozilla, dll.

Tanpa mengetahui dasar – dasarnya HTML seorang programmer web (lazim disebut web master) tidak akan memperoleh hasil yang maksimal, sebab walaupun software – software tersebut semakin menawarkan kemudahan dalam membuat halaman web, tetapi biasanya seseorang masih perlu untuk mengedit halaman web tersebut secara manual. Terutama untuk halaman web yang sangat kompleks. Terdapat beberapa HTML authoring (software yang digunakan untuk membuat atau mendesain halaman web), contohnya Adobe Dreamweaver, Adobe GoLive, MS FrontPage.

C. Peralatan Pendukung (*Tools System*)

Peralatan pendukung atau *tools system* yang penulis gunakan dalam pembuatan skripsi ini, diantaranya:

1. *HyperText Markup Language* (HTML)

Menurut Kadir (2009:12) “HTML adalah bahasa dengan tanda-tanda khusus yang digunakan di awal era *web* untuk menyajikan informasi”. HTML adalah singkatan dari *Hyper Text Markup Language* dan merupakan kumpulan dari *tag* atau tanda-tanda khusus untuk menampilkan halaman *web*. Dengan HTML kita bisa membuat halaman *web* kita sendiri. Struktur HTML dasar terdiri atas *head* dan *body* yang diapit oleh *tag* HTML yang ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kode HTML diawali dengan <html> dan diakhiri dengan </html>. Beberapa hal penting yang perlu diperhatikan dalam kode HTML adalah:

- a. Tanda < > menyatakan sebuah *tag*.
- b. Pada umumnya *tag* berpasangan. Contoh <html></html>, <head></head>, dan <body></body>.
- c. *Tag* yang tidak berpasangan antara lain adalah
 dan <hr />.
- d. Pada *tag* yang berpasangan, *tag* yang berkedudukan sebagai *tag* penutup mempunyai bentuk berupa </ >.

2. JavaScript

Menurut Kadir (2009:8) “JavaScript adalah bahasa skrip (bahasa yang kodenya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien”. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan

JavaScript dimungkinkan untuk memvalidasi masukan-masukan pada formulir sebelum formulir tersebut dikirim ke *server*.

3. CSS (*Cascading Style Sheet*)

Menurut Kadir (2009:450) menyebutkan bahwa “CSS (*Cascading Style Sheet*) digunakan dalam dokumen HTML untuk menciptakan suatu *style* (penyajian) yang dapat membuat kemampuan HTML menjadi lebih luas”. Yang menarik, *style* dapat didefinisikan pada *file* terpisah sehingga tidak menambah kerumitan dokumen dan *style* tersebut dapat digunakan pada sejumlah dokumen.

4. PHP (*PHP Hypertext Preprocessor*)

PHP merupakan bahasa pemrograman *web* yang bersifat *open source* dan telah banyak digunakan dalam membangun aplikasi *web*. Menurut Prasetyo (2005:2) “PHP (*PHP Hypertext Preprocessor*) merupakan bahasa yang mampu menghasilkan aplikasi *web* dinamis”. Dengan menggunakan bahasa pemrograman PHP kita dapat membuat halaman *web* menjadi lebih dinamis dan interaktif. PHP memberikan kita keleluasaan ketika menulis kode program, dengan catatan tetap mengacu pada ketentuan yang telah ditetapkan. PHP dapat disisipkan pada kode HTML atau dapat juga ditulis dalam kode PHP secara keseluruhan tergantung kebutuhan dan kebiasaan dari programmer yang membuatnya.

5. Basis Data (*Database*)

Menurut Shalahuddin dan Rosa A.S (2005:44) “Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data. Kebutuhan basis data dalam sistem informasi meliputi:

- a. Memasukan, menyimpan, dan mengambil data.
- b. Membuat laporan berdasarkan data yang telah disimpan.

6. *Database Management System (DBMS)*

Menurut Shalahuddin dan Rosa A.S (2011:45) “DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data”. Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DMBS. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

- a. Menyediakan fasilitas untuk mengelola akses data.
- b. Mampu menangani integritas data.
- c. Mampu menangani akses data.
- d. Mampu menangani *backup* data.

7. *Structure Query Language (SQL)*

Menurut Shalahuddin dan Rosa A.S (2011:46) “SQL adalah bahasa yang digunakan untuk mengelola data pada RDBMS”. SQL mulai berkembang pada tahun 1970an. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Pengaksesan data pada DBMS dengan SQL secara umum terdiri dari empat hal sebagai berikut:

- a. Memasukan data (*insert*).
- b. Mengubah data (*update*).
- c. Menghapus data (*delete*).
- d. Menampilkan data (*select*).

8. *Text Editor*

Text editor merupakan sebuah aplikasi yang digunakan untuk membuat atau mengedit *file text (plain text)*. *Text editor* dapat digunakan untuk membuat program komputer, mengedit *source code* bahasa pemrograman, serta membuat halaman *web*, atau *template web design*. Aplikasi ini secara umum digunakan untuk tujuan pemrograman bukan untuk pembuatan dokumen. Aplikasi semacam ini banyak tersedia di internet, mulai dari yang berbayar seperti *Dreamweaver* sampai yang gratis dan dapat digunakan dengan bebas seperti *Notepad++*.

D. *Unified Modelling Language (UML)*

Menurut Shalahuddin dan Rosa A.S (2011:118) “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. Menurut Sulistyorini (2009:23-29) dalam jurnal teknologi informasi yang berjudul Pemodelan Visual dengan Menggunakan UML dan Rational Rose menyebutkan bahwa “*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak”. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Menurut Munawar (2005:17) “UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek”. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang

sistem untuk membuat cetak biru atau visi dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain. UML dibangun atas model 4+1 *view*, Kelima *view* tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML, setiap *view* berhubungan dengan perspektif tertentu dimana sistem akan diuji. Kelima *view* tersebut yaitu:

1. *Use case view*, mendefinisikan perilaku eksternal sistem. *Use case view* memegang peranan khusus untuk mengintegrasikan *content* ke *view* yang lain.
2. *Design view*, mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di *use case*.
3. *Implementation view*, menjelaskan komponen-komponen fisik dari sistem yang akan dibangun.
4. *Process view*, berhubungan dengan hal-hal yang berkaitan dengan *concurrency* di dalam sistem.
5. *Deployment view*, menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan.

Didalam *UML* terdapat beberapa macam diagram yang dapat menggambarkan suatu sistem, berikut adalah beberapa diagram yang terdapat di dalam *UML*.

1. *Class Diagram*

Class dalam notasi UML digambarkan dengan kotak. “*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek” (Munawar, 2005b:42). *Class*

menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi).

Class mempunyai beberapa bagian yang menjelaskan isi dari *class*:

- a. *Attribute* adalah *property* dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada objek *class*. Sebuah *class* mungkin mempunyai nol atau lebih *attribute*.
- b. *Operation* adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau *class* yang lain dapat lakukan untuk sebuah *class*.
- c. *Responsibility* adalah keterangan tentang apa yang akan dilakukan *class* yaitu apa yang akan dicapai oleh *attribute* dan *operation*.

2. ***Sequence Diagram***

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* pesan yang diletakan diantara obyek-obyek ini dalam *use case*.

“*Sequence* diagram menggambarkan interaksi antar obyek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu” (Munawar, 2005c:87). *Sequence* diagram terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (obyek-obyek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Komponen utama *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Penjelasan dari komponen utama *sequence* diagram yaitu :

a. Obyek atau *Participant*

Obyek atau *participant* diletakan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operation dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.

b. *Message*

Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri.

c. *Time*

Time adalah diagram yang mewakili waktu pada arah *vertical*. Waktu dimulai dari atas kebawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibandingkan *message* yang lebih dekat.

3. ***Collaboration diagram***

“*Collaboration diagram* adalah perluasan dari obyek diagram. Obyek diagram menunjukkan obyek-obyek dan hubungannya satu dengan yang lain.

Collaboration diagram menunjukkan *message-message* obyek yang dikirimkan satu sama lain (Munawar, 2005d:101).

Collaboration diagram juga menggambarkan interaksi antar obyek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing obyek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor satu *Message* dari level yang sama memiliki *prefix* yang sama.

Dengan *collaboration diagram* memungkinkan untuk memodelkan pengiriman sebuah *message* ke banyak obyek pada *class* yang sama. Demikian juga halnya untuk menunjukkan adanya obyek aktif yang mengendalikan aliran *message*

4. Use Case Diagram

Use case diagram adalah penggambaran interaksi pengguna sistem (*actor*) dengan kasus (*use case*) yang telah disesuaikan dengan langkah-langkah (*scenario*). Diagram *use case* menunjukkan 3 aspek dari system yaitu *actor*, *use case* dan sistem, *sub sistem boundary*. *Actor* mewakili perang orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*.

5. Activity diagram

“*Activity Diagram* adalah teknik untuk mendiskripsikan *logika procedural*, proses bisnis dan aliran kerja dalam banyak kasus” (Munawar, 2005e:109). *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku *parallel* sedangkan *flowchart* tidak bisa.

Activity diagram tidak menunjukkan apa yang terjadi, tetap tidak menunjukkan siapa yang melakukan apa. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir, *Activity diagram* juga dapat menggambarkan proses *paralel* yang mungkin terjadi pada beberapa eksekusi. Simbol-simbol yang sering digunakan pada pembuatan *Activity diagram*.

6. ***Component diagram***

Component diagram merepresentasikan dunia nyata item yaitu *component software*. *Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. *Component software* adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

“Component diagram mengandung component, *interface* dan *relationship*”

(Munawar, 2005f:122).

7. ***Deployment diagram***

“*Deployment diagram* menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian *hardware*”
(Munawar, 2005g:125).

Deployment diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem terdiri dari *node-node* dimana setiap *node* diwakili untuk sebuah kubus. Garis yang menghubungkan antara dua kubus menunjukkan hubungan diantara kedua *node* tersebut. Tipe *node* bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor* (yang mengeksekusi component) atau *execution environment* (*software* yang menjadi *host* atau mengandung *software* yang lain).

8. ***Object Diagram***

Object diagram adalah diagram yang memberikan gambaran model instance-instance dari sebuah class. Diagram ini digunakan untuk menggambarkan sebuah sistem pada sebuah sudut pandang waktu tertentu.

Dengan menggunakan diagram ini anda dapat memeriksa keabsahan kelas-kelas diagram berikut aturan-aturan multiplisitasnya dengan “*real data*” dan mengujinya dengan scenario-skenario tertentu.

9. ***Statechart Diagram***

Statechart diagram, atau yang biasa juga disebut *state* diagram digunakan untuk mendokumentasikan beragam kondisi/keadaan yang bisa terjadi terhadap sebuah *class* dan kegiatan apa saja yang dapat merubah kondisi/keadaan tersebut. *State* diagram justru biasanya hanya memodelkan transisi yang terjadi hanya pada sebuah *class*. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

State, *Notasi State* menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya tumpul dengan nama state didalamnya.

Transition, Sebuah *Transition* menggambarkan sebuah perubahan kondisi objek yang disebabkan oleh sebuah event. *Transition* digambarkan dengan sebuah anak panah dengan nama *event* yang ditulis di atasnya, dibawahnya atau sepanjang anak panah tersebut.

Dalam pembuatan skripsi ini penulis hanya menggunakan empat diagram yaitu : *Use Case Diagram*, *Activity Diagram*, *Component Diagram*, *Deployment Diagram*, karena dalam pembuatan *website* tidak menggunakan *Class Diagram*.

E. *Entity Relationship Diagram (ERD)*

Model data adalah sekumpulan cara/peralatan/*tool* untuk mendeskripsikan data-data, hubungannya satu sama lain, semantiknya, serta batasan konsistensi.

Ada dua model data, yaitu : *Entity Relationship Diagram (ERD)* dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

Model ERD atau *Conceptual Data Model (CDM)* : model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu.

Model Relasional atau *Physical Data Model (PDM)* : model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara

data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki namayang unik.

Menurut Shalahuddin dan Rosa A.S (2011:50) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional”. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. Terdapat empat komponen utama yang biasa digunakan dalam menggambar ERD, yaitu:

1. *Entity* (entitas) adalah data inti yang akan disimpan, bakal tabel pada basis data.
2. *Relationship* (hubungan) adalah relasi yang menghubungkan antar entitas.
3. *Attribute* (atribut) adalah *field* atau kolom data yang butuh disimpan dalam suatu entitas.
4. *Association* (asosiasi) adalah penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

2.2 Penelitian Terkait

Penelitian sebelumnya yang berhubungan dengan penelitian ini antara lain sebagai berikut:

Menurut Dengan dkk (2009) “Pada hasil penelitian ini telah dikembangkan sebuah Sistem Informasi Akademik Berbasis *Web* dengan studi kasus pada SMP Negeri 4 Samarinda. Dimana dalam membangun sistem ini digunakan alat bantu pengembangan sistem yaitu *Data Flow Diagram (DFD)*, *Context Diagram*, *Entity Relationship Diagram (ERD)* dan *Flowchart* serta dengan menggunakan bahasa pemrograman PHP dan HTML dan MySQL sebagai *databasenya*”.

Menurut Prakoso dkk (2008) “*This academic application has functionalities to record student grade, subject data, basic competency and indicator,*

provide informations through News, give 2 way interaction media, storing students data, and calculating average score per class/student.”

Menurut Saraswati (2013) “Penelitian ini bertujuan untuk Menghasilkan Sistem Informasi Akademik yang mampu mengatasi kelemahan kelemahan dari sistem yang sedang dipakai saat ini, agar bermanfaat Untuk membantu para pegawai SMP Negeri 3 Pringkuku dalam pengolahan data dalam waktu yang relative lebih singkat. Metode pengumpulan data dilakukan dengan mengumpulkan data baik melalui wawancara maupun observasi. Sistem informasi akademik berbasis web ini dibangun dengan menggunakan bahasa pemrograman PHP dan menggunakan basis data MySql. Diharapkan Sistem ini dapat membantu memudahkan kinerja para pegawai pada SMP Negeri 3 Pringkuku”.