

BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media (Fricles Ariwisanto Sianturi; 2013: 43).

II.2. Uji Kompetensi

Berdasar pada arti estimologi kompetensi diartikan sebagai kemampuan yang dibutuhkan untuk melakukan atau melaksanakan pekerjaan yang dilandasi oleh pengetahuan, keterampilan dan sikap kerja. Sehingga dapatlah dirumuskan bahwa kompetensi diartikan sebagai kemampuan seseorang yang dapat terobservasi mencakup atas pengetahuan, keterampilan dan sikap kerja dalam menyelesaikan suatu pekerjaan atau tugas menyelesaikan suatu pekerjaan atau tugas sesuai dengan standar dengan standar performa yang ditetapkan.

Uji kompetensi merupakan evaluasi hasil belajar siswa selama belajar dan bisa di jadikan sebagai alat ukur keberhasilan siswa dan guru dalam melaksanakan

pembelajaran di sebuah sekolah. Awalnya kegiatan uji kompetensi ini akan dilaksanakan sebelum Ujian Nasional. Uji kompetensi adalah suatu sarana untuk menguji kemampuan siswa apakah siswa ini kompeten atau tidak kompeten di dalam mata pelajaran yang telah diberikan (Fahmy Syahputra; 2013:74).

II.3. *Wireless Application Protocol (WAP)*

Wireless Application Protocol (WAP) adalah sebuah protocol komunikasi dan suatu lingkungan aplikasi yang memungkinkan anda mengakses *internet* dan *web* dari *wireless handheld device*. *WAP* dirancang untuk bekerja dengan keanekaragaman selular jaringan transmisi data *wireless, handheld device*, dan sistem operasi. melihat atau membaca isi sebuah situs di *internet* dalam sebuah format teks khusus (Greace Widyastuti;2012:2).

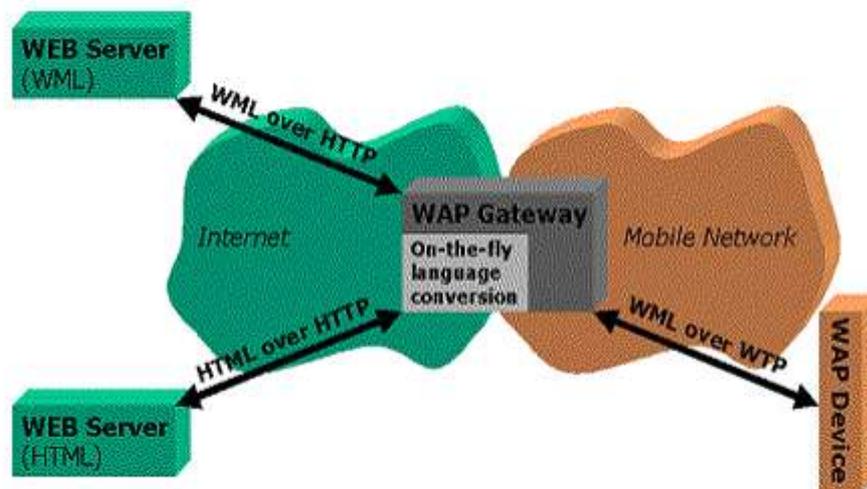
WAP adalah suatu *protocol* aplikasi yang memungkinkan internet dapat di akses oleh ponsel dan perangkat wireless lainnya. *WAP* membawa informasi secara online melalui internet langsung menuju ponsel atau klien *WAP* lainnya. Dengan adanya *WAP*, berbagai informasi dapat di akses setiap saat dengan menggunakan ponsel. *WAP* merupakan himpunan *protocol* yang di desain secara khusus untuk komunikasi internet dengan devais bergerak yang memiliki layar kecil dan bandwidth yang rendah. *WAP* juga dimaksudkan untuk meningkatkan fungsionalitas telepon genggam untuk layanan layanan real time, seperti informasi indeks saham, lalu-lintas, dan cuaca. Dengan *WAP* kita di beri peluang untuk membuat aplikasi-aplikasi dan layanan-layanan tersebut. Protokol-

protokol, standar, dan spesifikasi WAP di tentukan oleh suatu badan yang disebut WAP Forum (Mira Afrina; 2012:428) .

II.3.1. Cara Kerja WAP

Terdapat tiga bagian dalam akses WAP yaitu, perangkat wireles yang mendukung WAP, WAP gateway sebagai perantara, dan web server sebagai sumber Dokumen yang berada di dalam web server dapat berupa document html atau document wml. Document wml khusus ditampilkan melalui browser dari perangkat WAP. Sedangkan document html yang seharusnya akan ditampilkan melalui web browser, sebelum di baca melalui browser WAP diterjemahkan lebih dahulu oleh gateway agar dapat menyesuaikan dengan perangkat WAP. Jika seseorang pengguna ponsel menginginkan melihat suatu halaman web dengan format html, gateway akan menterjemahkan halaman tersebut ke dalam format wml. Namun, meskipun document Html bisa saja diakses lewat ponsel. Dokumen wml lebih disesuaikan dengan layanan ponsel yang kecil. Sehingga beberapa perusahaan telah mulai menyiapkan WAPsite disamping website yang telah ada.

Untuk menampilkan halaman WAP di butuhkan browser. Dalam bahasa ponsel, ini disebut *microbroser*. Seperti halnya mengetikan URL untuk mengakses website, kita juga akan melakukan hal yang sama untuk mengakses WAPsite di ponsel dengan mengakses web server melalui ISP dan login ke internet, maka halaman WAP akan di kirimkan dan dimunculkan dilayar ponsel. Bagi pengguna PC, juga disediakan browser *emulator* yang bisa di gunakan untuk mengakses situs ini (Mira Afrina; 2012:428).



Gambar II.1 Ilustrasi Kerja WAP

Sumber : (Mira Afrina; 2012)

II.4. PHP

Menurut kamus komputer, PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain, PHP adalah singkatan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada server (*server side HTML embedded scripting*) (Rulianto Kurniawan; 2010: 2).

II.5. MySQL

MySQL adalah salah satu jenis *database* server yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Management System*). MySQL ini mendukung bahasa pemrograman PHP. MySQL juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang simpel dan menggunakan *escape character* yang sama dengan PHP (Rulianto Kurniawan; 2010: 16).

II.6. UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah "bahasa" yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantic*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan. *Unified Modeling Language* biasa digunakan untuk :

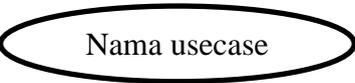
1. Menggambarkan batasan sistem dan fungsi – fungsi sistem secara umum, di buat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang di laksanakan secara umum, di buat dengan *interaction diagrams*.

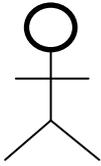
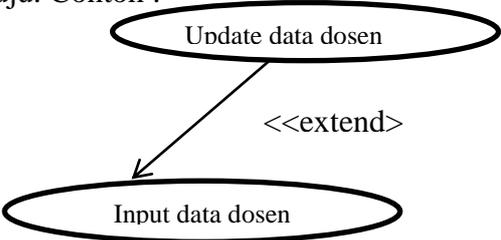
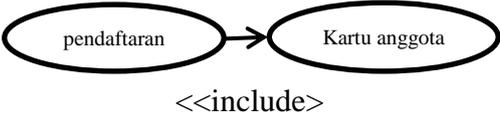
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Yuni Sugiarti; 2013 :36)

II.6.1. Use Case Diagram

Use case diagrams merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use case*, *actor* dan relasi. Berikut adalah simbol – simbol yang ada pada diagram *use case*. (Yuni Sugiarti; 2013: 42)

Tabel II.1 Simbol – simbol pada Use Case Diagram

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya ditanyakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .

<p>Aktor</p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Extend</p>  <p><<extend>></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, arah panah menunjuk pada <i>use case</i> yang dituju. Contoh :</p> 
<p>Include</p> <p><<include>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>, <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, contoh :</p> 

Sumber: (Yuni Sugiarti; 2013)

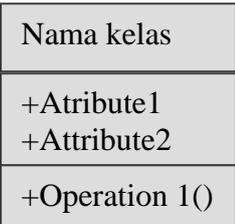
II.6.2. Class Diagram

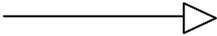
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang di sebut atribut dan metode atau operasi.

1. Atribut merupakan variabel- variabel yang di miliki oleh suatu kelas.
2. Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi – fungsi yang di miliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis – jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan – batasan yang terdapat dalam hubungan – hubungan objek tersebut. (Yuni Sugiarti; 2013: 57)

Tabel II.2 Simbol – simbol Class Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkus dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem

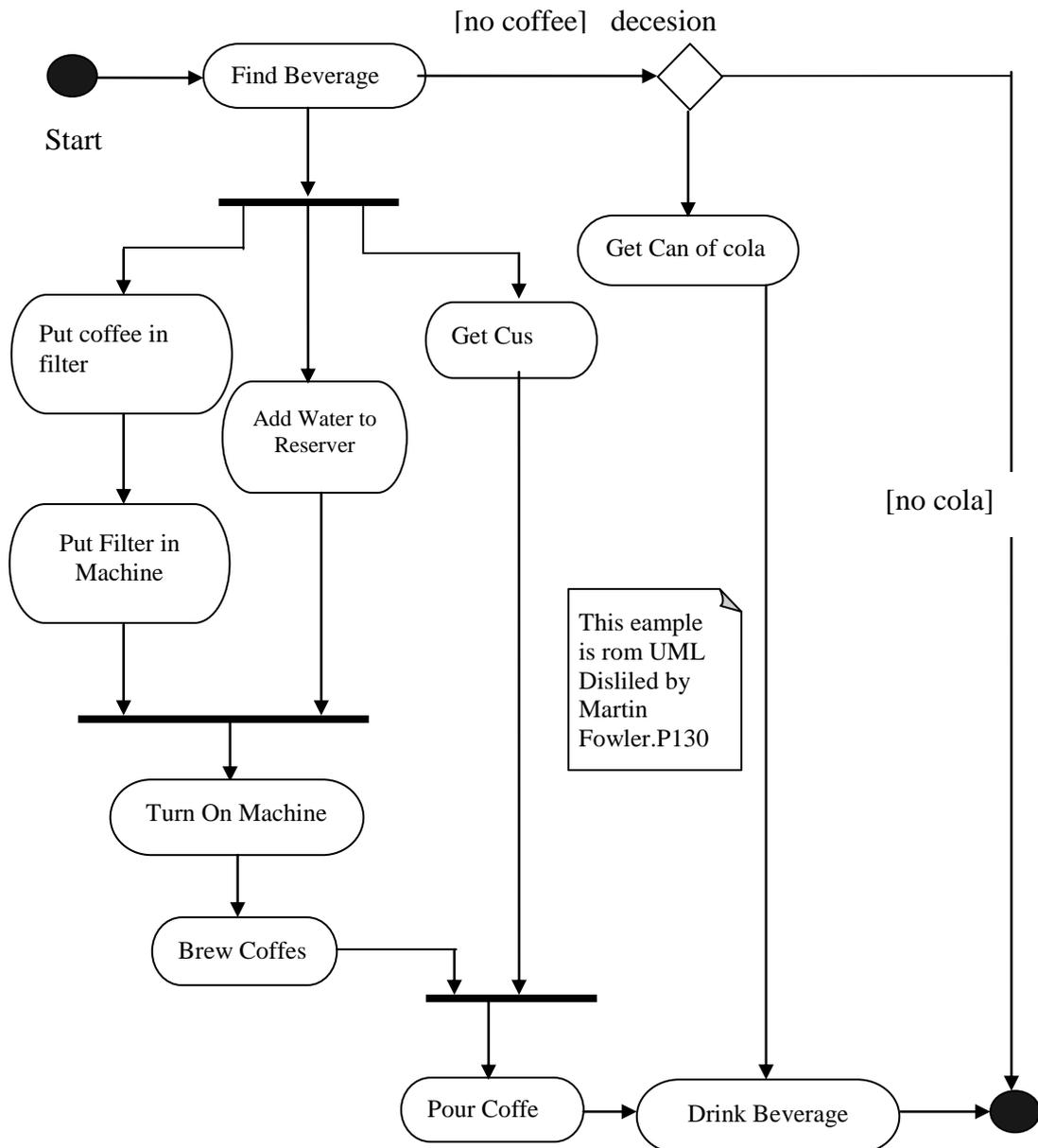
Antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi 1 _____ 1..*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu di gunakan oleh kelas yang lain, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus).
Kebergantungan / defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber : (Yuni Sugiarti ; 2013)

II.6.3. Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. *Activity* diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. (Yuni Sugiarti; 2013: 75)



Gamabar II.2 Activity Diagram

Sumber : (Yuni Sugiarti ; 2013)

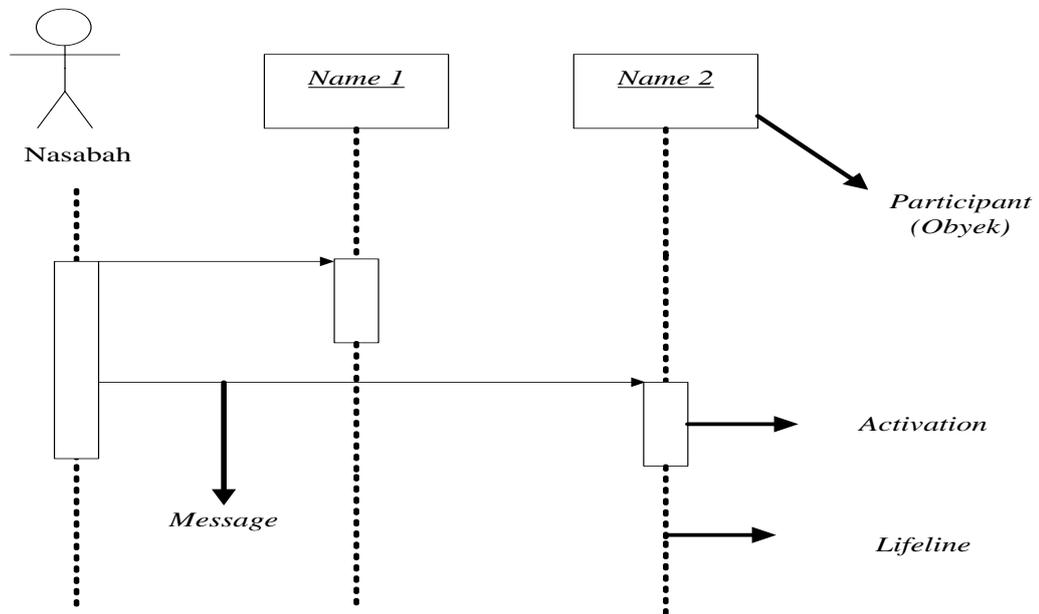
II.6.4. *Sequence Diagram*

Diagram *sequence* menggambarkan kelakuan/ pelaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Diagram *sequence* memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram *sequence* terdapat garis hidup objek. Garis hidup objek adalah garis *vertical* yang mencerminkan eksistensi sebuah objek sepanjang periode waktu. Sebagian besar objek – objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek – objek itu diletakkan dibagian atas diagram dengan garis hidup tergambar dari atas hingga bagian bawah diagram. Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan *destroy*, jika kasus ini terjadi, maka garis hidupnya juga berakhir.
2. Terdapat *focus* kendali (*Focus Of Control*), berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat. Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi. Pada diagram ini mungkin juga memperhatikan penyaringan (*nesting*) dan *focus* kendali yang disebabkan oleh proses rekursif dengan menumpuk *focus* kendali yang lain pada induknya. (Yuni Sugiarti; 2013: 70)

Berikut simbol – simbol yang ada pada sequence diagram.



Gambar II.3 Simbol Squence

Sumber : (Yuni Sugiarti ; 2013)

II.7. ERD (*Entity Relationship Diagram*)

Entity relationship diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas – entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan *struktur* basis data yang baik sehingga data dapat disimpan dan diambil secara *efisien*. Elemen – elemen diagram hubungan entitas yaitu :

1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau *abstrak* diman kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus dan buku.

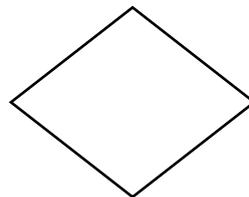


Gambar II.4 Simbol Entitas

Sumber : (Janner Simarmata,dkk; 2013)

2. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalny proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya mahasiswa bisa mengambil banyak mata kuliah.



Gambar II.5 Simbol Relasi

Sumber : (Janner Simarmata,dkk; 2013)

II.8. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional (www.utexas.edu). (Janner Simarmata & dkk; 2010 : 77)

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basis data menjadi tidak akurat, lambat, tidak *efisien*, serta tidak memberikan data yang diharapkan. Pada waktu menormalisasikan basis data, ada empat tujuan yang harus dicapai, yaitu:

1. Mengatur data dalam kelompok – kelompok sehingga masing – masing kelompok hanya mengenai bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basis data.
3. Membuat basis data yang datanya diakses dan dimanipulasi secara cepat dan *efisien* tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

II.9.1. Bentuk – Bentuk Normalisasi

1. Bentuk normal pertama (1NF)

Tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap *atribut* harus mempunyai nilai data yang *atomic* (bersifat *atomic value*).

2. Bentuk normal kedua (2NF)

Normal kedua didasari atas konsep *full functional dependency* (ketergantungan fungsional sepenuhnya) yang dapat didefinisikan sebagai berikut.

Jika A dan B adalah *atribut-atribut* dari suatu relasi, B dikatakan *full function dependency* (miliki ketergantungan fungsional sepenuhnya) terhadap A, jika B adalah tergantung fungsional terhadap A, tetapi tidak secara tepat memiliki ketergantungan fungsional dari *subset* (himpunan bagian) dari A.

3. Bentuk normal ketiga (3NF)

Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data di dalam *database* tersebut akan *inkonsisten*/tidak teratur. Anomali *update* ini disebabkan oleh suatu ketergantungan transitif (*transitive dependency*). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3-NF).

4. Bentuk normal *boyce-code* (BCNF)

Suatu relasi dalam basis data harus dirancang sedemikian rupa sehingga mereka memiliki ketergantungan sebagian (*partial dependency*), maupun ketergantungan transitif (Janner Simarmata & dkk; 2010 : 77).

II.10. Kamus Data

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data biasanya berisi:

1. Nama - nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data. (Rosa A.S & M Shalauddin; 2011 : 67)