

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Kedua kelompok definisi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya.

Pembahasan tentang sistem sebenarnya bukanlah sesuatu hal yang baru. Memang di dunia ini tidak ada yang sama sekali baru. Kalau ada sesuatu yang baru, sebenarnya hal itu pun sebenarnya sudah lama ada. Dinilai baru karena baru ditemukan, baru diungkap, atau baru diketahui oleh orang banyak. Untuk sampai pada kesepakatan di antara orang-orang terhadap sesuatu yang tampak baru itu, terlebih dahulu terjadi pertentangan pendapat yang bisa berlanjut ke perdebatan. Perdebatan ini menghasilkan suatu keputusan, yang seolah-olah baru, walau pada hakikatnya bukanlah hal yang baru (Tata Sutabri ; 2012 : 2).

II.1.1. Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu (Yakub; 2012 : 1).

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri ; 2012 : 6).

II.1.2. Elemen-elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem yaitu tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan.

a. Tujuan

Tujuan ini menjadi motivasi yang mengarahkan pada sistem, karena tanpa tujuan yang jelas sistem menjadi tak terarah dan tak terkendali.

b. Masukan (*input*) sistem

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud maupun yang tidak berwujud. Masukan berwujud adalah bahan mentah, sedangkan yang tidak berwujud adalah informasi.

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.

d. Keluaran (*output*)

Keluaran (*output*) merupakan hasil dari pemrosesan sistem dan keluaran dapat menjadi masukan untuk subsistem lain.

e. Batas (*boundary*) sistem

Batas (*boundary*) sistem adalah pemisah antara sistem dan daerah diluar sistem. Batas sistem menentukan konfigurasi, ruang lingkup atau kemampuan sistem

f. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik (*feedback*), sedangkan umpan balik ini digunakan untuk mengendalikan masukan maupun proses. Tujuannya untuk mengatur agar sistem berjalan sesuai dengan tujuan.

g. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem.

II.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang. Klasifikasi sistem tersebut diantaranya yaitu sistem abstrak (*abstract system*), sistem fisik (*physical system*), sistem tertentu (*deterministic system*), sistem tak tentu (*probabilistic system*), sistem tertutup (*close system*) dan sistem terbuka (*open system*).

- a. Sistem tak tentu (*probabilistic system*), adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsure probabilitas.

Sistem arisan merupakan contoh *probabilistic system* karena sistem arisan tidak dapat diprediksi dengan pasti.

- b. Sistem abstrak (*abstract system*), adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem teologia yang berisi gagasan tentang hubungan manusia dengan Tuhan merupakan contoh *abstract system*.
- c. Sistem fisik (*physical system*), adalah sistem yang ada secara fisik. Sistem komputer, sistem akuntansi, sistem produksi, sistem sekolah dan sistem transportasi merupakan contoh *physical system*.
- d. Sistem tertentu (*determinic system*), adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Sistem komputer sudah diprogramkan merupakan contoh *determinic system* karena program komputer dapat diprediksi dengan pasti.
- e. Sistem tertutup (*close system*), sistem yang tidak bertukar materi, informasi atau energi dengan lingkungan. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan, misalnya : reaksi kimia dalam tabung yang terisolasi.
- f. Sistem terbuka (*open system*), adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Sistem perdagangan merupakan contoh *open system* karena dapat dipengaruhi oleh lingkungan.

II.2. Konsep Dasar Informasi

Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan. Informasi dapat berupa data mentah, data tersusun, kapasitas sebuah saluran informasi, dan sebagainya (Yakub; 2012 : 8).

II.3. Konsep Dasar Sistem Informasi

Sistem Informasi merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi juga dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk menyajikan informasi. Sistem informasi merupakan sistem pembangkit informasi, kemudian dengan integrasi yang dimiliki antar subsistem, maka sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya. Sistem informasi juga merupakan suatu kumpulan dari komponen-komponen dalam organisasi yang berhubungan dengan proses

penciptaan dan aliran informasi. Pada lingkungan berbasis komputer, sistem informasi menggunakan perangkat keras dan lunak komputer, jaringan telekomunikasi, manajemen basis data, dan berbagai bentuk teknologi informasi yang lain dengan tujuan untuk mengubah sumber data menjadi berbagai macam informasi yang dibutuhkan oleh pemakai (Yakub; 2012 : 17-18).

II.4. *Software Development Life Cycle (SDLC)*

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses pengembangan atau mengubah sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (bersadarkan *best practice* atau cara-cara yang sudah teruji baik).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

1. Inisiasi (*Initiation*)
2. Pengembangan Konsep Sistem (*System Concept Development*)
3. Perencanaan (*Planning*)
4. Analisis Kebutuhan (*Requirements Analysis*)
5. Desain (*Design*)
6. Pengembangan (*Development*)
7. Intergrasi dan Pengujian (*Integration and Test*)
8. Implementasi (*Implementation*)
9. Operasi dan Pemeliharaan (*Opertaion and Maintenance*)

10. Disposisi (*Disposition*)

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya antara lain sebagai berikut :

1. Model *Waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berturut-turut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*).

2. Model Prototipe

Dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembnag perangkat lunak.

3. Model RAD (*Rapid Application Development*)

Adalah model proses pengembag perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. (Rosa A.S, M.Shalahiddin ; 2011 : 24-32)

II.5. Analisis

Analisis sistem adalah kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan yang tidak bagus, dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru. Hal

tersebut terlihat sederhana, namun sebenarnya tidak. Banyak hambatan yang akan ditemui dalam proses tersebut (Rosa A.S dan M. Shalahuddin; 2011: 16).

II.6. Pengendalian Internal

Pengendalian Internal (*Internal Control*) adalah suatu proses yang dipengaruhi oleh dewan direksi entitas, manajemen dan personel lainnya, yang dirancang untuk memberikan kepastian yang beralasan terkait dengan pencapaian sasaran kategori sebagai berikut : efektivitas dan efisiensi operasi; keandalan pelaporan keuangan dan ketaatan terhadap hukum dan peraturan yang berlaku (Dasaratha V. Rama / Frederick L. Jones ; 2009 : 132).

Sistem pengendalian intern meliputi struktur organisasi, metode dan ukuran-ukuran yang dikoordinasikan untuk menjaga kekayaan organisasi, mendorong efisiensi dan mendorong dipatuhinya kebijakan manajemen. Definisi sistem pengendalian intern tersebut menekankan tujuan yang hendak dicapai dan bukan pada unsur-unsur yang membentuk sistem tersebut. Dengan demikian, pengertian pengendalian intern tersebut diatas berlaku baik dalam perusahaan yang mengolah informasinya secara manual, dengan mesin pembukuan maupun dengan komputer (Mulyadi ; 2008 : 163).

II.7. Piutang Dagang

Prosedur pencatatan piutang bertujuan untuk mencatat mutasi piutang usaha perusahaan kepada setiap debitur. Mutasi piutang adalah disebabkan oleh

transaksi penjualan kredit, penerimaan kas dari debitur, dan prosedur distribusi penjualan.

Informasi mengenai piutang yang dilaporkan kepada pihak manajemen adalah:

1. Saldo piutang pada saat tertentu kepada setiap debitur
2. Riwayat pelunasan piutang yang dilakukan oleh setiap debitur
3. Umur piutang kepada setiap debitur pada saat tertentu (Mulyadi ; 2008 : 257-261)

II.8. Basis Data

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (*key*) dari tiap *file* yang ada. Dalam satu file terdapat *record-record* sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan dan menunjukkan dalam satu pengertian yang lengkap dalam satu *record*. Berdasarkan pengertian di atas dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu : bersifat *data oriented* dan bukan *program oriented*, dapat digunakan oleh beberapa program aplikasi tanpa perlu

mengubah basis datanya. Hal ini juga dapat dikembangkan dengan mudah baik *volume* maupun strukturnya sehingga dapat memenuhi kebutuhan sistem-sistem baru secara mudah. Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Adapun tujuan basis data diantaranya sebagai efisiensi yang meliputi *speed, space & accuracy*, menangani data dalam jumlah besar, kebersamaan pemakaian (*sharebility*), dan meniadakan duplikasi dan inkonsistensi data (Yakub; 2012 : 51-53).

II.9. DataBase Management System (DBMS)

DataBase Management System (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk membuat dan mengelola basis data. DBMS berisi suatu koleksi data dan satu set program untuk mengakses data. DBMS merupakan perangkat lunak (*software*) yang menentukan bagaimana data tersebut diorganisasi, disimpan, diubah, dan diambil kembali. Perangkat lunak ini juga menerapkan mekanisme pengamanan data, pemakaian data bersama, dan konsistensi data (Yakub; 2012 : 55).

Perangkat lunak yang termasuk DBMS (*DataBase Management System*) adalah sebagai berikut:

Tabel II.1. Nama DBMS

Nama DBMS	Nama Perusahaan
Access	Microsoft Corporation
DB2	IBM
Informix	IBM

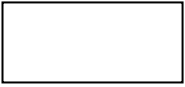
Ingres	Computer Associate
MySQL	The MySQL AB Company
Oracle	Oracle Corporation
PostgreSQL	www.postgresql.com
Sybase	Sybase Inc

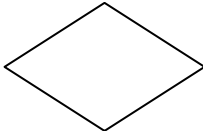
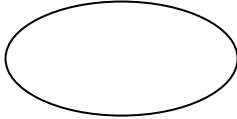

(Sumber : Yakub; 2012 : 55)

II.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data (*database*). Model data ini juga akan membantu pada saat melakukan analisis dan perancangan basis data, karena model data ini akan menunjukkan bermacam-macam data yang dibutuhkan dan hubungan antardata. ERD ini juga merupakan model konseptual yang dapat mendeskripsikan hubungan antara *file* yang digunakan untuk memodelkan struktur data serta hubungan antar data.

Tabel II.2. Simbol ERD

Simbol	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.

	<p>Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain ; satu ke satu, satu ke banyak, dan banyak ke banyak.</p>
	<p>Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.</p>
	<p>Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasinya</p>

(Sumber : Yakub; 2012 : 60)

ERD terbagi atas tiga komponen, yaitu entitas (*entity*), atribut (*attribute*), dan relasi atau hubungan (*relation*). Secara garis besar entitas merupakan dasar yang terlibat dalam sistem. Atribut atau *field* berperan sebagai penjelas dari entitas, dan relasi atau hubungan menunjukkan hubungan yang terjadi antara dua entitas (Yakub; 2012 : 60)

II.11. Normalisasi

Normalisasi (*normalize*) merupakan salah satu cara pendekatan atau teknik yang digunakan dalam membangun desain logik basis data *relation* dengan menerapkan sejumlah aturan dan kriteria standar. Tujuan dari normalisasi adalah untuk menghasilkan struktur tabel yang normal atau baik. Teknik normalisasi adalah upaya agar desain logik dari tabel-tabel berada dalam bentuk normal yang dapat didefinisikan dengan menggunakan ketergantungan fungsi (*functional dependency*). (Yakub; 2012 : 70)

II.11.1. Bentuk Normalisasi

Bentuk normal adalah suatu aturan yang dikenakan pada relasi-relasi atau tabel-tabel dalam basis data dan harus dipenuhi oleh relasi atau tabel tersebut pada level-level normalisasi. Suatu relasi dikatakan dalam bentuk normal tertentu jika memenuhi kondisi tertentu juga. Beberapa bentuk normalisasi diantaranya adalah bentuk tidak normal (*unnormalize*), normalisasi pertama, normalisasi kedua, dan normalisasi ketiga (Yakub; 2012 : 71)

II.11.2. Tidak Normal

Bentuk tidak normal merupakan kumpulan data yang direkam dan tidak ada keharusan dengan mengikuti suatu format tertentu. Pada bentuk tidak normal terdapat repeating group sehingga pada kondisi seperti ini akan menjadi permasalahan dalam melakukan manipulasi data (*insert, update, dan delete anomalies*). (Yakub; 2012 : 71)

II.11.3. Normalisasi Pertama

Dalam *relational database* tidak diperkenankan adanya repeating group karena dapat berdampak terjadinya anomalies. Oleh karena itu tahap unnormalisasi akan menghasilkan bentuk normal pertama. Normalisasi ke satu, suatu relasi atau tabel memenuhi normal ke satu jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal (scalar value) dalam satu baris atau record. (Yakub; 2012 : 71)

II.11.4. Normalisasi Kedua

Dalam perancangan basis data *relational* tidak diperkenankan adanya *partial functional dependency* kepada *primary key*, karena dapat berdampak terjadinya *anomalies*. Oleh karena itu tahap normalisasi pertama akan menghasilkan bentuk normal kedua yang dapat didefinisikan sebagai berikut : normalisasi kedua, suatu relasi memenuhi relasi kedua jika dan hanya jika relasi tersebut memenuhi normal pertama dan setiap atribut yang bukan kunci (*non key*) bergantung secara fungsional secara utuh kepada kunci utama (*primary key*) (Yakub; 2012 : 72).

II.11.5. Normalisasi Ketiga

Dalam perancangan basis data relasional tidak diperkenankan adanya *transitive dependency* karena dapat berdampak terjadinya *anomalies*. Oleh karena itu harus dilakukan normalisasi tahap ketiga yang dapat didefinisikan sebagai berikut : normalisasi ke tiga, suatu relasi memenuhi normal ketiga jika dan hanya jika relasi tersebut memenuhi normal ke dua dan setiap atribut yang bukan kunci (*non key*) tidak mempunyai *transitive dunctional dependency* kepada kunci utama (*primary key*) (Yakub; 2012 : 72).

II.12. Kamus Data

Kamus data (*data dictionary*) digunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukkan (*input*) dan

keluaran atau output dapat dipahami secara umum (memiliki standar penulisan). (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

(Rosa A.S, M. Shalahuddin; 2011: 67-68)

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut (Lihat tabel II.3)

Tabel II.3. Simbol-Simbol Dalam Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[I]	Baikatau...
{ } ⁿ	N kali diulang / bernilai banyak
()	Data opsional
**	Batas komentar

(Sumber : Rosa A.S, M. Shalahuddin; 2011: 68)

II.13. UML (*Unified Modeling Language*)

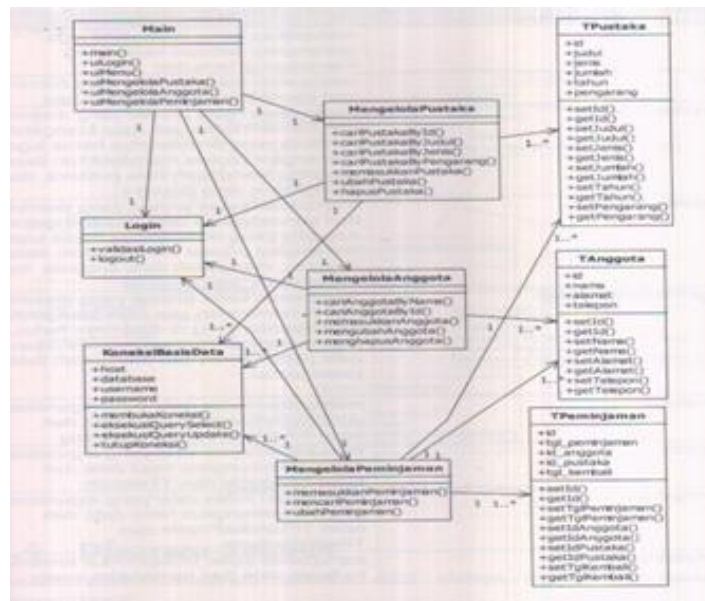
UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini.

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram yang lain menyebutkan delapan jenis diagram karena ada beberapa diagram yang digabung, misalnya 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 Kelas

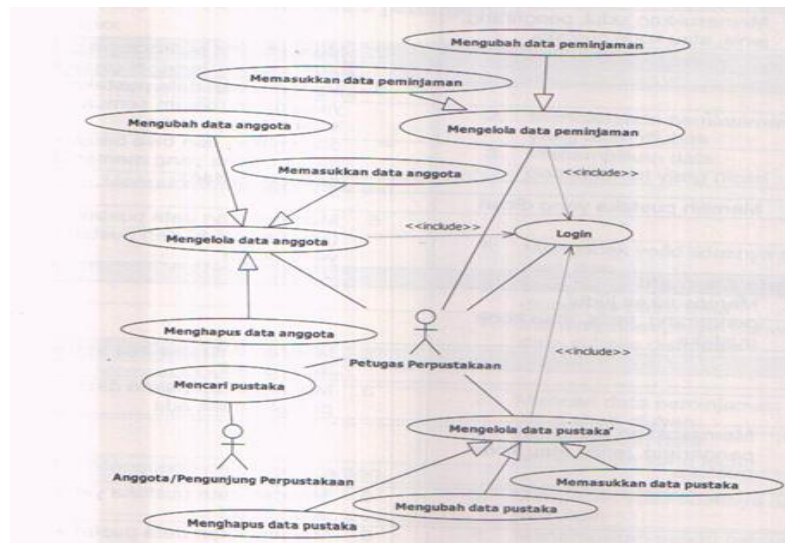
Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada permodelan sistem berorientasi objek. Meskipun bersifat statis, seringpula diagram kelas memuat kelas-kelas aktif.



Gambar II.1. Contoh Class Diagram
(Sumber : Rosa. A.S, M. Shalahuddin ; 2011 : 161)

2. Diagram Use Case

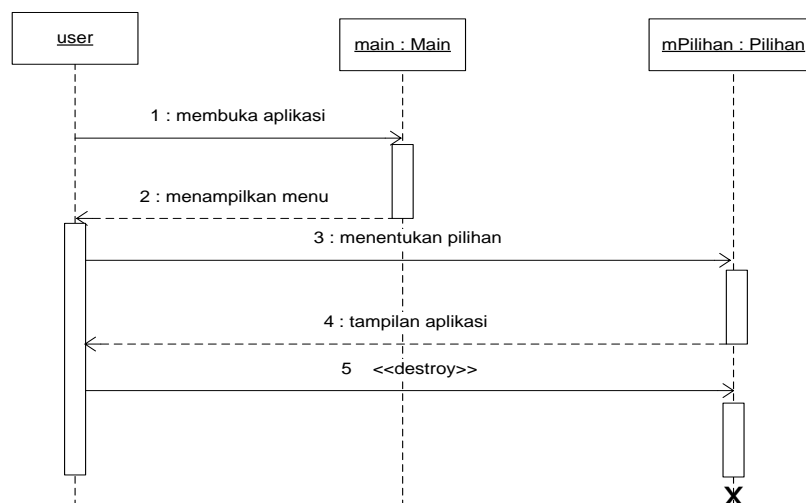
Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. (Lihat gambar II.3)



Gambar II.3. Contoh Use Case Diagram
 (Sumber : Rosa A.S, M. Shalahuddin ; 2011 : 160)

3. Diagram Interaksi dan *Sequence* (urutan)

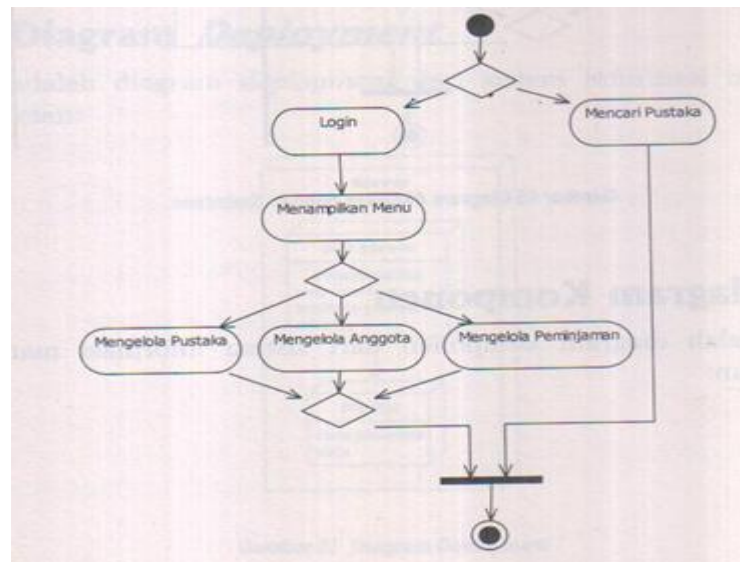
Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.



Gambar II.4. Contoh Sequence Diagram
 (Sumber : Rosa A.S, M. Shalahuddin; 2011: 164)

4. Diagram Aktivitas (*Activity Diagram*)

Bersifat dinamis. Diagram aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari satu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.



Gambar II.7. Contoh Activity Diagram
(Sumber : Rosa A.S, M. Shalahuddin ; 2011 : 177)

Keempat ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semua dibuat sesuai dengan kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya (Misalnya *Data Flow Diagram*, *Entity Relationship Diagram*, dan sebagainya) (Prabowo Pudjo Widodo dan Herawati ; 2011 : 6-12)

II.14. Visual Basic 2010

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Basic 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C #*, dan *Visual F#*.

Semua IDE pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun database standar yang disertakan adalah *Microsoft SQL Server 2008 express*.

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat di dalamnya antara lain dukungan terhadap *library* terbaru *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap pengembangan aplikasi menggunakan *Microsoft SilverLight*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*.

Bahasa *Visual Basic 2010* sendiri awalnya berasal dari bahasa pemrograman yang sangat populer di kalangan *programmer computer*, yaitu bahasa *Basic*, yang oleh *Microsoft* diadaptasi dalam program *Microsoft Quick Basic*, seiring dengan perkembangan teknologi komputasi dan desain, *Microsoft* mengeluarkan produk yang dinamakan *Microsoft Visual Studio*, yang didalamnya termasuk *Visual Basic 2010* (Wahana Komputer ; 2010 : 2-3)

II.15. SQL Server 2008 R2

Microsoft SQL Server 2008 R2 adalah data *platform* yang paling canggih, terpercaya, dan terukur yang pernah dirilis hingga saat ini. Dibangun pada keberhasilan rilis *SQL Server 2008* yang asli, *SQL Server 2008 R2* telah memberi dampak pada organisasi di seluruh dunia dengan kemampuan inovatifnya, memberdayakan pengguna akhir melalui *intelijen bisnis self-service* (BI), memperkuat efisiensi dan kolaborasi antara *database administrator* (DBA) dan pengembang aplikasi, dan skala untuk mengakomodasi beban kerja data yang paling menuntut.

Sekarang, lebih dari sebelumnya, organisasi memerlukan database *platform* yang dapat dipercaya, hemat biaya, dan terukur yang menawarkan efisiensi dan dikelola secara *self-service* BI. Wajah organisasi-organisasi selalu berubah sesuai kondisi bisnis dalam ekonomi global, kendala anggaran IT, dan kebutuhan untuk tetap kompetitif dengan mendapatkan dan memanfaatkan informasi yang tepat pada waktu yang tepat. Dengan *SQL Server 2008 R2*, mereka dapat mengatasi tekanan untuk mencapai ini. Rilis ini memberikan sebuah platform database kelas *enterprise* pemenang penghargaan dengan kemampuan yang kuat yang meningkatkan efisiensi melalui pemanfaatan sumber daya yang lebih baik, pemberdayaan pengguna akhir, dan *scaling* keluar dengan biaya lebih rendah. Perangkat tambahan untuk skalabilitas dan kinerja, ketersediaan tinggi, keamanan perusahaan, pengelolaan perusahaan, data *warehouse*, pelaporan, *self-service* BI, kolaborasi, dan integrasi ketat dengan *Microsoft Visual Studio 2010*,

Microsoft SharePoint 2010, dan *SQL Server PowerPivot* untuk *SharePoint* membuat *platform database* terbaik yang tersedia. *SQL Server 2008 R2* dianggap versi minor *upgrade* dari *SQL Server 2008*. Namun, untuk *upgrade* kecil menawarkan sejumlah terobosan besar terbaru, kemampuan yang DBA dapat dimanfaatkan.

Microsoft telah membuat investasi besar dalam produk *SQL Server* secara keseluruhan, namun, fitur-fitur baru dan kemampuan terobosan yang harus paling menarik DBA adalah kemajuan dalam aplikasi dan administrasi *multi-server*. Bagian ini memperkenalkan beberapa fitur-fitur baru dan kemampuan kelompok produk *SQL Server* telah melakukan investasi yang cukup besar dalam meningkatkan aplikasi dan kemampuan manajemen *multi-server*. Beberapa aplikasi utama dan perangkat tambahan administrasi *multi-server* yang memungkinkan organisasi untuk lebih baik dalam mengelola lingkungan *SQL Server* mereka masukkan, yakni :

a. *SQL Server Utilitas*

Ini adalah fitur baru yang digunakan untuk memonitor pengelolaan terpusat dan mengelola aplikasi database dan contoh *SQL Server* dari satu antarmuka manajemen dikenal sebagai *Control Point Utility (UCP)*. Contoh dari *SQL Server*, aplikasi *data-tier*, *file database*, dan *volume* dikelola dan dilihat dalam *SQL Server Utility*.

b. *Utility Control Point (UCP)*

Sebagai titik penalaran pusat untuk *SQL Server Utility*, *Control Point Utilitas* mengumpulkan konfigurasi dan kinerja informasi dari kasus

dikelolanya *SQL Server* setiap 15 menit. Setelah data telah dikumpulkan dari kasus yang dikelola, *SQL Server Utilitas dashboard* dan sudut pandang dalam *SQL Server Management Studio (SSMS)* menyediakan DBA dengan ringkasan kesehatan *SQL Server* sumber daya melalui evaluasi kebijakan dan analisis sejarah.

c. *Aplikasi data-tier*

Sebuah aplikasi *data-tier* (DAC) adalah satu unit penyebaran yang berisi semua skema database, dan persyaratan penyebaran digunakan oleh aplikasi. DAC A dapat digunakan dalam salah satu dari dua cara: bisa ditulis dengan menggunakan Server proyek aplikasi *data-tier* *SQL* dalam *Visual Studio 2010*, atau dapat dibuat dengan mengekstraksi definisi DAC dari *database* yang sudah ada dengan *Ekstrak Wisaya Application Data-Tier* di *SSMS*. Melalui penggunaan DAC, penyebaran aplikasi data dan kolaborasi antara pengembang data tier dan DBA meningkat secara signifikan.

d. *Utilitas Explorer dashboard*

Dashboard dalam *SQL Server Utilitas* menawarkan DBA wawasan yang luar biasa dalam pemanfaatan sumber daya dan negara kesehatan untuk kasus dikelola yang *SQL Server* dan disebarkan aplikasi *data-tier* seluruh perusahaan. Sebelum pengenalan dari *SQL Server Utility*, DBA tidak memiliki alat yang ampuh disertakan dengan *SQL Server* untuk membantu mereka dalam memantau pemanfaatan sumber daya (Ross Mistry, Stacia Misner ; 2010 : 3-11).