BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Dengan berbagai pendekatan, beragam pula istilah "sistem" didefinisikan., Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling bergantung dan terintegrasi dalam kesatuan variabel atau komponen. Terdapat dua kelompok pendekatan sistem, yaitu menekankan pada prosedur dan komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat dari tahapan-tahapan instruksi. Sedangkan pendekatan yang menekankan pada komponen mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu *"Serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi"* (Riyanto, dkk; 2009 : 22).

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Asbon Hendra : 2012 ; 157).

II.2. Sistem Pendukung Keputusan

Sistem penunjang keputusan (SPK) adalah suatu sistem informasi untuk membantu manajer level menengah untuk proses pengambilan keputusan setengah terstruktur supaya lebih efektif dengan menggunakan model-model analisis dan data yang tersedia (Tata Sutabri : 2013 : 61).

II.3. Metode *Fuzzy*

Orang yang belum pernah mengenal logika *fuzzy* pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang amat rumit dan tidak menyenangkan. Namun, sekali seseorang mulai mengenalnya, ia pasti akan sangat tertarik dan akan menjadi pendatang baru untuk ikut serta mempelajari logika *fuzzy*. Logika *fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *fuzzy* modern dan metodis baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada pada diri kita sejak lama (Rika Rosnelly ; 2012: 63).

Terdapat beberapa metode yang sering digunakan oleh peneliti untuk membangun suatu Sistem Penentuan Keputusan untuk menentukan Tentor. Salah satunya adalah Metode *Fuzzy*. Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Intelegent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Konsep logika *fuzzy* pertama sekali diperkenalkan oleh Professor Lotfi A. Zadeh dari Universitas California, pada bulan Juni 1965. Logika *fuzzy* merupakan generalisasi dari logika klasik yang hanya memiliki dua nilai keanggotaan, yaitu 0

dan 1. Dalam logika *fuzzy*, kebenaran suatu pernyataan berkisar dari sepenuhnya besar sampai dengan sepenuhnya salah.Dengan landasan diatas, penulis ingin sekali mengangkat topik diatas menjadi judul Skripsi Penulis dengan Judul "Sistem Penunjang Keputusan Pemilihan Tentor Pada Lembaga Pendidikan Komputer *Century* Menggunakan Metode *Fuzzy* (Khodijah Rein, dkk, 1).

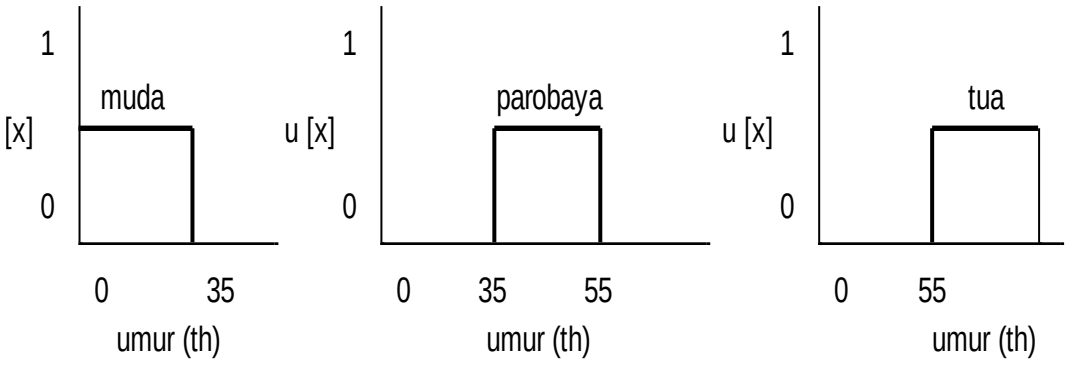
Menurut Rika Rosnelly (2012 : 63) Misalkan variabel umur dibagi menjadi 3 kategori, yaitu :

Muda umur < 35 tahun

Parobaya 35 ≤ umur ≥ 55 tahun

Tua umur 55 tahun

Nilai keanggotaan secara grafis, himpunan muda, parobaya, dan tua ini dapat dilihat pada gambar II. 1 berikut ini :



Gambar II. 1. Fuzzy Logic

II.4. Unified Modelling Language

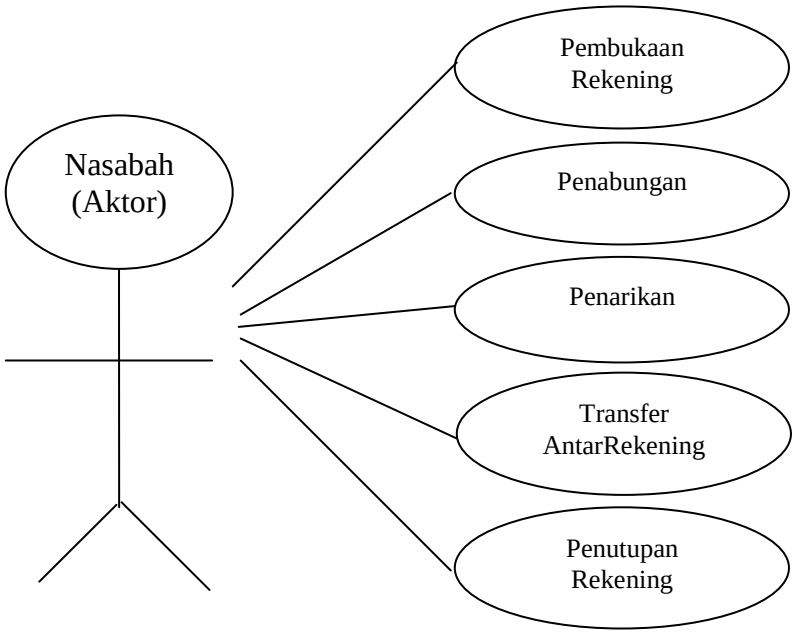
Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan. Adapun tujuan pemodelan (dalam kerangka pengembangan sistem/perangkat lunak aplikasi) adalah sebagai sarana visualisasi dan komunikasi antar anggota tim pengembang (saat seorang analis/perancang sistem/perangkat lunak bekerja dalam tim yang beranggotakan beberapa/banyak anggota) serta sebagai sarana dokumentasi (bermanfaat untuk menelaah perilaku sistem secara seksama dan bermanfaat untuk menguji (*testing*) sistem yang telah selesai dikembangkan).

Dalam hal ini kita sebagai perancang sistem/perangkat lunak kita menggambarkan komponen-komponen sistem/perangkat lunak dalam bentuk-bentuk geometri tertentu, misalnya untuk menggambarkan suatu kelas (*class*) dalam aplikasi kita membentuk empat persegi panjang, untuk menggambarkan hubungan antarkelas kita menggunakan garis lurus (Adi Nugroho ; 2009 : 5).

II.4.1. Use Case Diagram

Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem untuk berusaha tahu secara pasti mengenai hal yang menjadi kebutuhan dan harapan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya. Selanjutnya, *use case diagram* tidak hanya sangat penting pada saat analisis, tetapi juga sangat penting dalam tahap perancangan (*design*), untuk mencari kelas-kelas yang terlibat alam aplikasi, dan untuk melakukan pengujian (*testing*).

Saat akan mengembangkan *use case diagram,* hal yang pertama kali harus dilakukan adalah mengenali *actor* untuk sistem yang sedang dikembangkan. Dalam hal ini, ada beberapa karakteristik untuk para *actor*, yaitu *actor* yang ada di luar sistem yang sedang dikembangkan dan *actor* yang berinteraksi dengan sistem yang sedang dikembangkan. (Adi Nugroho ; 2009 : 7)



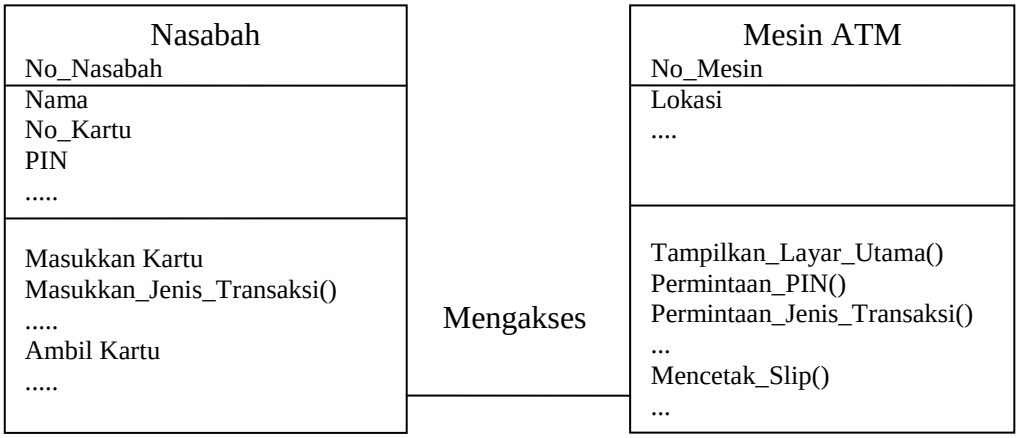
Gambar II.2. Contoh Use Case Diagram
(Adi Nugroho ; 2009 : 8)

II.4.2. Class Diagram

Class didefenisikan sebagai kumpulan/himpunan objek yang memiliki kesamaan dalam atribut/properti, perilaku (operasi), serta cara berhubungan dengain objek lain.

Selain itu, kita juga mendefenisikan objek sebagai konsep, abstraksi dari sesuatu dengan batas nyata, sehingga kita dapat menggambarkan secara sistematis. Pemahaman objek memiliki dua fungsi, yaitu :

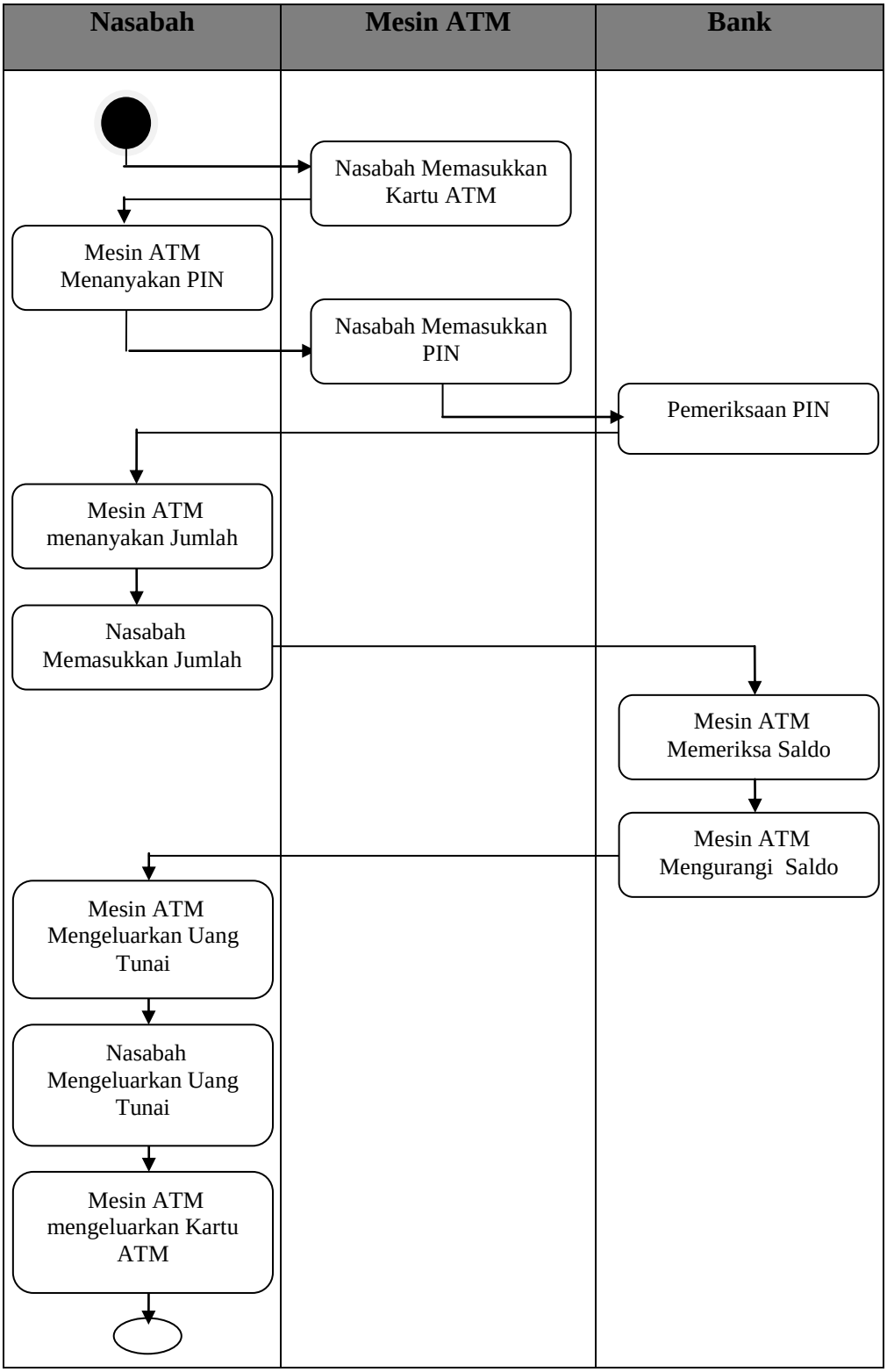
a. Memudahkan untuk mempelajari secara seksama hal-hal yang ada di dunia nyata.
b. Menyediakan suatu dasar yang kuat dalam implementasi ke dalam sistem terkomputerisasi (Adi Nugroho ; 2009 :17-18).



Gambar II.3. Contoh Class Diagram
(Nugroho ; 2009: 39)

II.4.3. Activity Diagram

Apakah langkah yang harus kita lakukan selanjutnya setelah kita membuat use case diagram ? use case diagram merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci. Oleh karena itu, kita harus memperinci lagi perilaku sistem untuk masing-masing use case yang ada. Apa perkakas (tool) yang bisa kita gunakan ? jika kasus kita cukup sederhana, mungkin kita bisa menggunakan skenario seperti yang tercantum berikut, sementara jika kasusnya cukup kompleks, kita mungkin bisa menggunakan activity diagram agar bisa mendapatkan gambaran yang lebih menyeluruh (Adi Nugroho ; 2009 : 10).



Gambar II.4. Contoh Activity Diagram
(Adi Nugroho ; 2009 : 11)

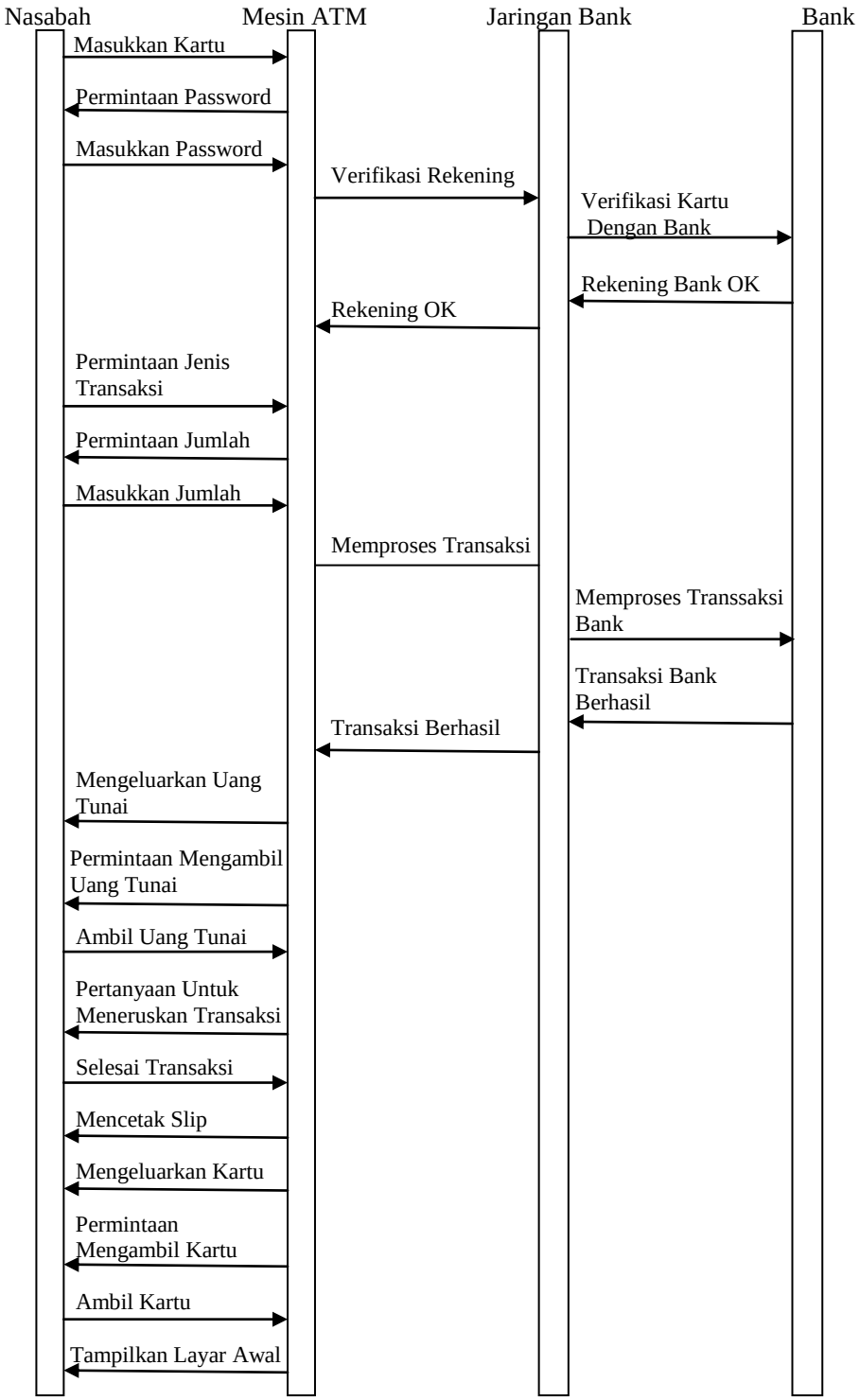
II.4.4. Sequence Diagram

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*.

Jadi dengan kata lain sekuensial diagram menunjukkan aliran fungsionalitas berdasarkan urutan waktu serta kejadian yang nantinya akan menetukan metode/fungsi atribut masing-masing. Dimana fungsi-fungsi tersebut akan diterapkan pada suatu kelas/objek.

Perhatikan gambar II.4. dimana terlihat pengelompokkan *event-event* serta fungsi masing-masing atribut tersebut. Di dalam diagram terlihat jelas bagaimana aliran suatu proses kejadian dimana seorang nasabah yang akan melakukan transaksi dengan sebuah mesin ATM. Dari diagram tersebut kita mengetahui *event-event* yang terjadi, seperti : Nasabah memasukkan kartu ATM, Mesin ATM merespon dengan meminta *password* atau PIN, dan selanjutnya.

Kita dapat melihat setiap fungsi atribut dan *event-event* apa saja yang terjadi. Sehingga melalui diagram sekuensial ini kita dapat merancang suatu program aplikasi yang baik, sehingga dalam menghadapi sebuah kasus yang benar-benar kompleks diagram sekuensial ini sangat membantu **(**Adi Nugroho ; 2009 : 35).



Gambar II.5. Contoh Sequence Diagram
(Adi Nugroho ; 2009 : 36)

II.5. *Database* Dan *ERD*

Desain database merupakan pekerjaan yang penting dalam pembuatan atau pengembangan sistem, karena desain *database* akan mendapatkan susunan data atau *table* yang efektif dan efisien. Alat desain *database* yang popular ada dua, yaitu : ERD (*Entity Relationship Diagram*) dan Normalisasi. Jika memakai *Normalisasi* harus mendapatkan Data Dasar (Dokumen Dasar), sedangkan ERD tidak perlu. Dalam desain ERD terbagi dua tahapan yaitu: *Preliminary* Desain (Disain Awal) dan *Final Design* (Disain Akhir). Tetapi disain Akhir dari ERD juga berisi Normalisasi (Yuniar Supardi : 2008 : 9)

II.6. Data *Dictionary* (kamus data)

Dalam suatu rancangan *database*, data *dictionary* digunakan untuk menjelaskan atau mendeskripsikan kolom-kolom pada masing-masing tabel yang akan dibuat ke dalam *database*. Deskripsi kolom yang dimaksud di sini meliputi tipe data, lebar karakter atau digit, serta keterangan tentang kunci relasi (Budi Raharjo : 2011 ; 59).

Tabel II.1. Tabel kategori

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Kategori_id	INT	11	NOT NULL	Primary Key
Kategori_nama	VARCHAR	25		

Tabel II.2. Tabel pengarang

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Pengarang_id	CHAR	3	NOT NULL	Primary Key
Pengarang _nama	VARCHAR	30		

Tabel II.3. Tabel penerbit

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Penerbit _id	CHAR	4	NOT NULL	Primary Key
Penerbit _nama	VARCHAR	50		

Tabel II.4. Tabel buku

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key
Buku_judul	VARCHAR	75		
Penerbit_id	CHAR	4		
Buku_tglterbit	DATE	-		
Buku_jmlhalaman	INT	11		
Buku_deskripsi	TEXT	-		
Buku_harga	DECIMAL	10,0		

Tabel II.5. Tabel link_buku_pengarang

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key dan Forign Key
Pengarang_id	CHAR	3	NOT NULL	Primary Key dan Forign Key

Tabel II.6. Tabel link_buku_kategori

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key dan Forign Key
kategori_id	CHAR	11	NOT NULL	Primary Key dan Forign Key

(Sumber : Budi Raharjo : 2011 : 59)

II.7. Normalisasi

Menurut Yuniar Supardi (2008 : 10) tahapan normalisasi terdiri dari beberapa bentuk, yaitu:

1. Bentuk Tak Normal (UNF / *Un Normal Form*).
2. Bentuk Normal Pertama (1 NF / *First Normal Form*).

   Bentuk Normal pertama memiliki ciri: Data berbentuk *file-file* (*file* datar), *record* disusun sesuai kedatangan, masih mungkin terjadi penyimpangan data (anomali data). Anomali data dapat berupa insert

anomali, delete anomali, update, anomali, dan redundancy data (data duplikat).

3. Bentuk Normal Kedua (2 NF / *Second Normal Form*).

   Bentuk Normal kedua memiliki ciri; Tidak terjadi anomali data, setiap *field*/atribut bukan kunci harus tergantung fungsi (*Functional Depedency*) terhadap *field*/atribut kunci, masih mungkin terjadi *transitive dependency* (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu *table*). Model objek mencapai bentuk normal kedua, sehingga penulis mendesain mulai bentuk normal ketiga dan bentuk normal boyce codd. Sedangkan untuk bentuk tak normal sudah dari dokumen dasar berupa Faktur, Nota, dan laporan *Stock of Name*.

4. Bentuk Normal Ketiga (3 NF / *Third Normal Form*).

   Table yang memenuhi Bentuk Normal Ketiga harus tidak terdapat *Transitive Depedency*. Bentuk normal ketiga dari sistem *inventory* sebagai berikut:

Tabel II.7. 3NF

Pelanggan

*Kode Pelanggan	Nama_Pelanggan	Alt_Pelanggan	Tlp_Pelanggan

Transj

*No_Faktur	Qty_Jual	Harga_Jual	**Kode_Barang

Faktur

*No_Faktur	Tgl_Jual	Total_Jual	Pembuat	Penerima	**Kode_ Pelanggan	**No_SO

Barang

*No_Barang	Nama_Barang	Harga_Beli	Harga_Jual	Quantity

(Yuniar Supardi ; 2008 : 11)

5. Bentuk Normal *Boyce Codd* (BCNF / *Boyce Codd Normal Form*). Karena tak ada *field* bukan kunci tergantung secara parsial (bagian) kunci dalam satu tabel, maka bentuk normal ketiga juga merupakan bentuk BCNF.
6. Normal yang lebih tinggi.

II.8. Microsoft Visual Basic

Visual basic merupakan salah satu bahasa pemograman yang andal dan banyak digunakan oleh pengembang untukmembangun berbagai macam aplikasi *windows*. *Visual basic 2008* merupakan aplikasi pemograman yang menggunakan teknologi. *NET Framework 3.5*. Teknologi. *NET Framework 3.5* merupakan komponen *windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi, dan halaman *web*. Teknologi .*Net Framework 3.5* mempunyai 2 komponen utama, yaitu *CLR (Common Language Runtime)* dan *Class Library*. *CLR* digunakan untuk menjalankan aplikasi yang berbasis.*NET*, sedangkan Library adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi (Wahana Komputer;2010:2).

II.9. Microsoft SQL Server

Bahasa query merupakan bahasa khusu yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (query) yang berhubungan dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya. Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL merupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data relasional (RDBMS) (Ema Utami dan Anggi Dwi Hartanto : 2012 : 63).

II.10. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan DSS (*Decision Support System*) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk mem-bantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan. Aplikasi DSS menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran

pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan kepu-tusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia (Deni Harahap, dkk : 1-2).

II.11. Metode *Fuzzy*

Terdapat beberapa metode yang sering digunakan oleh peneliti untuk membangun suatu Sistem Penentuan Keputusan untuk menentukan Tentor. Salah satunya adalah Metode *Fuzzy*. Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Intelegent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Konsep logika *fuzzy* pertama sekali diperkenalkan oleh Professor Lotfi A. Zadeh dari Universitas California, pada bulan Juni 1965. Logika *fuzzy* merupakan generalisasi dari logika klasik yang hanya memiliki dua nilai keanggotaan, yaitu 0 dan 1. Dalam logika *fuzzy*, kebenaran suatu pernyataan berkisar dari sepenuhnya besar sampai dengan sepenuhnya salah.Dengan landasan diatas, penulis ingin sekali mengangkat topik diatas menjadi judul Skripsi Penulis dengan Judul "Sistem Penunjang Keputusan Pemilihan Tentor Pada Lembaga Pendidikan Komputer *Century* Menggunakan Metode *Fuzzy* (Khodijah Rein, dkk, 1).