

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian yang berjudul. *Sistem Pendukung Keputusan Seleksi Penerima Beras Untuk Keluarga Miskin*. Langkah-langkah dalam merancang sistem yang meliputi analisa sistem secara umum dan terperinci dengan menggunakan alat bantu *Flowchart* sistem dan *Entity Relational Diagram* (ERD), metode direct, menggunakan database MYSQL, bahasa pemrograman Java. Dengan menggunakan beberapa kriteria untuk menentukan keluarga miskin diantaranya: pekerjaan tidak tetap, penghasilan maksimal 15.000,00/hari, orang tua tidak ada menanggung, jumlah keluarga minimal 3 orang dari masing-masing keluarga, kemudian dari masing-masing kriteria yang telah ditentukan tersebut diberikan bobot penilaian. Interval bobot yang dipakai dalam penilaian keluarga ini adalah 0-4. Semakin tinggi nilai bobot penilaian, maka semakin tinggi pula nilai intensitas total penilaian keluarga. Hasil keluarga yang berhak menerima beras (jumlah nilai kriteria ≥ 12). Kelemahan, inputan kriteria penilaian terlalu sedikit, membuat hasil output kurang valid [3].

Pada Penelitian yang penulis lakukan ini berjudul, "*Sistem Pendukung Keputusan Seleksi Penerimaan bantuan beras untuk warga miskin (RASKIN)*". Penelitian ini menggunakan paradigma siklus hidup pengembangan sistem, dengan menggunakan Rekayasa Perangkat Lunak model *Waterfall*. Rekayasa Perangkat Lunak pendekatan sistematis dalam analisis, perancangan, implementasi meliputi tahapan pengumpulan data dengan observasi, wawancara dan studi pustaka. Analisis sistem dengan menggunakan pemodelan AHP. Perancangan sistem menggunakan, relasi antar tabel, diagram konteks, dan diagram aliran data. Bahasa program yang digunakan adalah *Visual Basic* dengan *Sql*, dan kriteria yang digunakan sebagai berikut penghasilan, pekerjaan, jumlah keluarga bersekolah, luas bangunan, jenis dinding, jenis lantai dan kepemilikan aset, dari masing-masing keluarga akan di perhitungkan bobot kriterianya dengan

menggunakan metode AHP. Jika nilai warga diatas 0,60 maka warga tersebut layak menerima beras miskin.

Penelitian selanjutnya dari Sri Kusumadewi dan Hari purnomo(2010: 3). Sistem pendukung Keputusan adalah bagian dari sistem informasi yang berbasis komputer yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semiterstruktur yang spesifik.

Berbeda dari penelitian yang sudah ada sebelumnya, penelitian yang akan penulis ambil adalah membuat sistem pendukung keputusan(SPK) untuk menentukan warga yang berhak mendapatkan RASKIN di Desa Petekeyan, Kecamatan Tahunan, Kabupaten Jepara dengan menggunakan Visual Basic sebagai bahasa programnya dan Sql sebagai databasenya. Sistem ini berupa desktop yang didalamnya memuat penilaian terhadap kriteri-kriteria dari hasil penilaian yang diharapkan dapat menentukan warga yang berhak mendapatkan RASKIN dengan tepat.

2.2 Landasan Teori

2.2.1 Beras Miskin (RASKIN)

Raskin adalah program dari pemerintah untuk mengurangi beban pengeluaran dari rumah tangga miskin sebagai bentuk dukungan dalam meningkatkan ketahanan pangan dengan memberikan perlindungan sosial beras murah dengan jumlah maksimal 15kg/rumah tangga setiap bulan sekali dengan masing-masing seharga Rp. 1600,00-/kg di titik distribusi. Program ini mencakup di seluruh provinsi [6].

2.2.2 Sistem Pendukung Keputusan

2.2.2.1 Pengertian Sistem Pendukung Keputusan

Konsep Sistem Pendukung keputusan (SPK) / *Decisison Support System*(DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S, Scot Morton dengan istilah *Management Decision System*. Sistem Pendukung Keputusan adalah suatu sistem yang berbasis komputer yang ditujukan untuk

membantu pengambil keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan.

Menurut Little, DSS adalah sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan. Sedangkan Bonzeck, dkk mendefinisikan DSS sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi yaitu [2]:

1. Sistem bahasa merupakan mekanisme untuk memberikan komunikasi antara pengguna dan komponen DSS lain.
2. Sistem pengetahuan adalah repositori pengetahuan domain masalah yang ada pada DSS baik sebagai data atau sebagai prosedur. Repositori pengetahuan merupakan lokasi penyimpanan pengetahuan yang sebenarnya dalam sistem manajemen pengetahuan. Serupa dengan database, tetapi umumnya berorientasi teks.
3. Sistem pemrosesan masalah merupakan hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan.

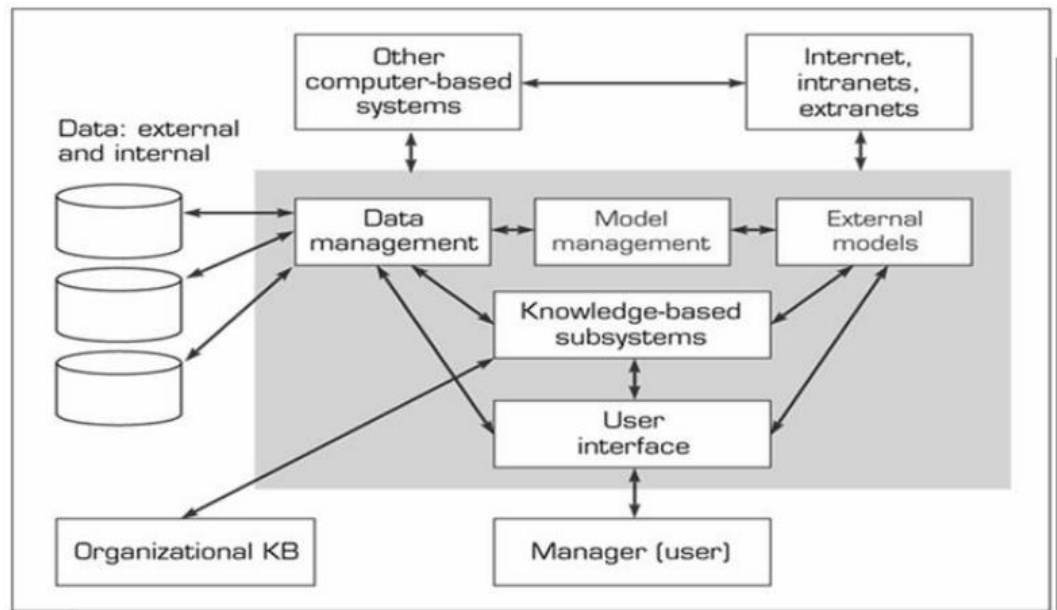
2.2.2.2 Tujuan Sistem Pendukung Keputusan

Tujuan Sistem Pendukung Keputusan yang dikemukakan oleh Keen dan Scott dalam McLeod mempunyai tiga tujuan yang akan dicapai adalah:

1. Membantu manajer membuat keputusan untuk memecahkan masalah semistruktur.
2. Mendukung penilaian manajer bukan mencoba menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajer daripada efisiensinya.

Apabila SPK dapat terdiri dari beberapa subsistem, yaitu: subsistem manajemen data, subsistem manajemen model dan subsistem antarmuka pengguna. Selain itu SPK juga bisa memiliki subsistem manajemen basis-pengetahuan yang mendukung subsistem-subsistem lainnya.

2.2.2.3 Komponen Sistem Pendukung Keputusan



Gambar 2.1 Skematik SPK

(Sumber : Turban)

Aplikasi SPK dibentuk dari subsistem-subsistem. Adapun komponen-komponen dari Sistem Pendukung Keputusan sebagai berikut[2]:

1. Subsistem Manajemen Data

Subsistem yang pertama adalah *Data Management Subsystem*, meliputi database yang berisi data yang relevan terhadap situasi yang bersangkutan, dan diatur oleh *software* yang disebut *Database Management system (DBMS)*. *Data Management Subsystem* yang terhubung otomatis dengan visual basic dimana data-data inputan dari visual basic yang dapat diintegrasikan dengan data warehouse perusahaan, suatu penyimpanan data-data yang relevan untuk pembuatan keputusan perusahaan.

2. Subsistem manajemen Model

Model Management Subsystem merupakan *software* yang meliputi model keuangan, statistik, ilmu manajemen, dan kuantitatif lainnya, yang menyediakan kemampuan analisis sistem dan manajemen *software* yang tepat. *Software* ini juga sering disebut *Model Bas Management System (MBMS)*. Komponen ini dapat diintegrasikan dengan penyimpanan *external* model perusahaan.

3. Subsistem Manajemen Berbasis Pengetahuan

Subsistem ini dapat mendukung subsistem lainnya atau berperan sebagai komponen yang bebas. Subsistem ini dapat diintegrasikan dengan *knowledge depository* perusahaan yang disebut *organizational knowledge base*.

4. Subsistem Antarmuka Pengguna

User berkomunikasi dan memberi perintah pada SPK melalui subsistem ini. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari SPK berasal dari interaksi yang intensif antara komputer dan pembuat keputusan.

2.2.2.4 Proses Pengambilan Keputusan

Proses pengambilan keputusan meliputi tiga tahapan utama yaitu tahap intelegensi, desain dan pemilihan. Namun kemudian ditambahkan dengan tahap keempat yaitu tahap implementasi. Keempat tahapan tersebut dapat dijelaskan sebagai berikut:

1. Tahap Penelusuran (*Intelegence*)

Tahap penelusuran merupakan tahap pendefinisian masalah serta identifikasi informasi yang dibutuhkan berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil. Langkah ini sangat penting karena sebelum suatu tindakan diambil, tentunya persoalan yang dihadapi harus dirumuskan secara jelas terlebih dahulu.

2. Perancangan (*Design*)

Tahap perencanaan merupakan tahap analisa dalam kaitan mencari atau merumuskan alternatif-alternatif pemecahan masalah. Setelah permasalahan dirumuskan dengan baik, maka tahap berikutnya adalah merancang atau membangun model pemecahan masalahnya dan menyusun berbagai alternatif pemecahan masalah.

3. Pemilihan (*Choice*)

Dengan mengacu pada rumusan tujuan serta hasil yang diharapkan, selanjutnya manajemen memilih alternatif solusi yang diperkirakan paling sesuai. Pemilihan alternatif ini akan mudah dilakukan kalau hasil yang diinginkan terukur atau memiliki nilai kuantitas tertentu.

4. Implementasi (Implementation)

Tahap implementasi merupakan tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan-perbaikan.

2.3 Analytical Hierarchy Process (AHP)

Menurut Kusriani [2], Analytical Hierarchy Process (AHP) adalah sebuah model dengan hirarki fungsional dengan input utamanya berupa persepsi manusia. Keberadaan hirarki memungkinkan dipecahkannya masalah kompleks dan tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hirarki. Pengambilan keputusan dimulai dengan membuat layout dari keseluruhan hirarki keputusannya. Hirarki tersebut menunjukkan faktor-faktor yang ditimbang serta sebagai alternatif yang ada. Kemudian sejumlah perbandingan berpasangan dilakukan untuk mendapatkan penetapan nilai faktor dan evaluasinya. Sebelum penetapan dilakukan, terlebih dahulu ditentukan kelayakan hasil nilai faktor yang didapat dengan mengukur tingkat konsistensinya. Pada akhir alternatif dengan jumlah nilai tertinggi dipilih sebagai alternatif terbaik.

1. Prinsip dasar AHP

Menurut Kusriani [2] dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah:

a. Membuat *Hirarki*

Sistem yang kompleks bisa dipahami dengan menjadi elemen-elemen pendukung, menyusun elemen secara *hirarki*, dan menggabungkannya atau mensintesisnya.

b. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty dalam Kusriani [2], untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan saaty bisa diukur menggunakan tabel analisis seperti ditunjukkan pada tabel 2.1.

Tabel 2.1 Nilai Skala Perbandingan Berpasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	

c. Menentukan prioritas

Untuk setiap kriteia dan alternatif, perlu dilakukan perbandingan berpasangan(pairwise comparisons). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

d. Konsistensi Logis

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

2.3.1 Prosedur AHP

Pada dasarnya, prosedur atau langkah-langkah dalam metode AHP meliputi:

1. Mendefinisikan masalah dan menentukan solusi yang di inginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan Prioritas Elemen
Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan. Matrik perbandingan berpasangan diisi menggunakan bilangan untuk mempersentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.
3. Sintesis
Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :
 - a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
 - b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
 - c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.
4. Mengukur konsistensi
Dalam pembuatan keputusan penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :
 - a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya
 - b. Jumlahkan setiap baris
 - c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan

d. Jumlahkan hasil bagi di atas dengan banyaknya elemen yang ada, hasilnya disebut λ maks

5. Hitung Consistency index (CI) dengan rumus

$$CI = (\lambda \text{ maks} - n)/n$$

6. Hitung Consistency Ratio (CR) dengan rumus

$$CR = CI/IR$$

Dimana CR = Consistency Ratio

CI = Consistency Index

IR = Index Random Consistency

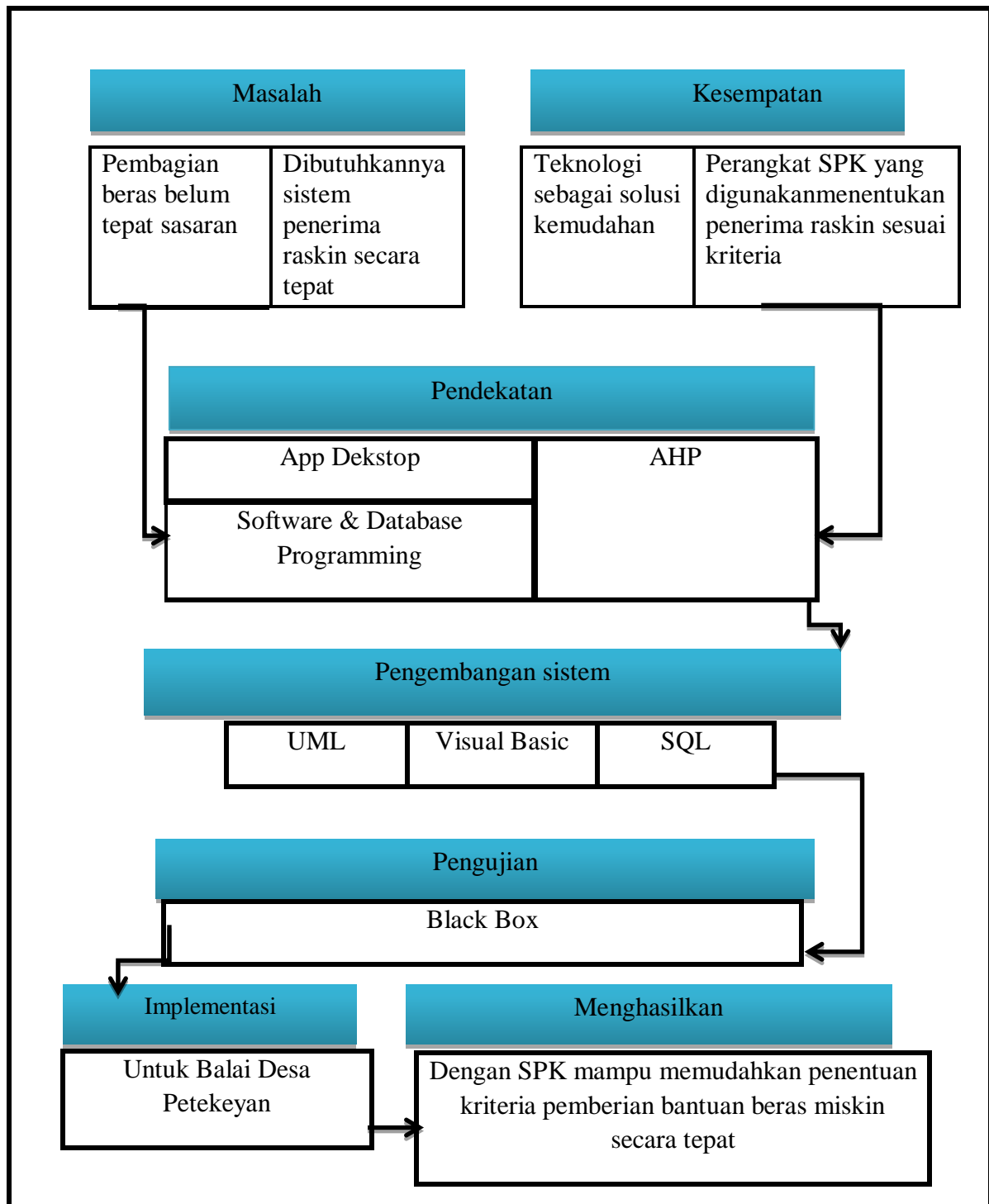
7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10 %, maka penilain data judgement harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar

Datar Indeks Random konsistensi (IR) bisa dilihat dalam tabel 2.2 dibawah ini :

Tabel 2.2. Indeks Random Konsistensi

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

2.4 KERANGKA PEMIKIRAN



Gambar 2.2. Kerangka Pemikiran Pembagian Beras Miskin

Kerangka pikir di atas menggambarkan tentang masalah yang ada dan peluang yang dimiliki untuk menyelesaikan masalah yang ada. Kemudian

dibangun aplikasi pengelolaan nilai yang dirancang dengan *software* dan *database programming*. Aplikasi ini kemudian diimplementasikan pada Balaidesa dan hasilnya adalah berupa alternatif dalam menentukan warga penerima raskin.

2.5 Perancangan Sistem

2.5.1 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak [5].

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C[4].

1. Tujuan UML

- a. Memberikan model yang siap pakai, bahasa pemodelan visual ekspresif.
- b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek-praktek terbaik yang terdapat dalam bahasa pemodelan.

2. Diagram Grafis UML


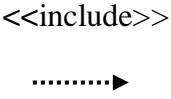
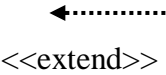

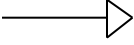

a. Use Case Diagram

Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem[5]. Use case mendeskripsikan interaksi tipikal antara pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

Bahasan *use case*, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem. Aktor dapat meliputi pelanggan, petugas layanan konsumen, manajer penjualan dan analis produk. Aktor-aktor tersebut menggunakan *use case*. Seorang aktor dapat menggunakan banyak *use case*, sebaliknya

sebuah *use case* juga dapat digunakan beberapa aktor. Aktor tidak harus manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem lain tersebut merupakan aktor.

Table 2.1 Notasi Use Case Diagram

No.	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2		Include	Kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya
3		Extend	Kelakuan yang hanya berjalan di bawah kondisi tertentu
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya
5		Generalization	Hubungan generalisasidan spesialisasi dari umum-khusus antara dua use case atau lebih dimana use case yang satu lebih umum dari yang lainnya.
6		Use Case	Gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun


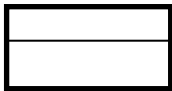

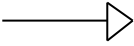
b. Class Diagram

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *ClassDiagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek

tersebut[5]. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah class.

ClassDiagram membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama proses analisis, *Class Diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *Class Diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut dan metode.

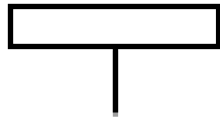
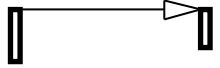
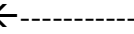
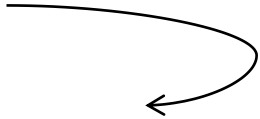


Tabel 2.2 Notasi Class Diagram

No.	Gambar	Nama	Keterangan
1		Asosiasi Berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
2		Class	Himpunan objek-objek yang berbagi atribut serta operasi yang sama
3		Association	Apa yang menghubungkan antara objek satu dengan objek yang lainnya
4		Generalization	Hubungan generalisasidan spesialisasi dari umum-khusus antara dua use case atau lebih dimana use case yang satu lebih umum dari yang lainnya.

c. Sequence Diagram

Sequence Diagram merupakan interaksi onyek yang disusun dalam suatu urutan waktu. *Sequence Diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case* [5]. Tipe diagram ini sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaan dan mudah untuk dimengerti.






Tabel 2.3 Notasi Sequence Diagram

No	Gambar	Nama	Keterangan
1		Object	Objek <i>entity</i> , antarmuka yang saling berinteraksi
2		Object Message	Komunikasi antar object yang berupa data
3		Return Message	Komunikasi antar object yang berupa pesan
4		Message To self	Komunikasi ke diri sendiri
5		Timeline	Sebuah kotak yang digambarkan di bawah aktor dan sistem untuk menggambarkan aliran waktu
6		Actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat

d. Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang, 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[5]. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktifitas ke keadaan sesaat (*state*).

Table 2.4 Notasi Activity Diagram

No.	Gambar	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berintraksi satu sama lain
2		State transition	State dari sistem yang menggambarkan alur sebuah activity
3		Initial Node	Bagaimana objek dibentuk atau diawali
4		Activity Finally Node	Bagaimana objek dibentuk dan dihancurkan
5		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran



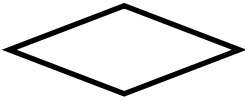

e. Rasional Rose

Menurut Flower, M., *Rasional Rose* adalah *tool* pemodelan visual untuk pengembangan sistem berbasis objek yang handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem[5]. *Rasional rose* mendukung pemodelan bisnis yang membantu para pengembang memahami sistem secara komprehensif. *Rasional rose* juga membantu analisis sistem dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna.

2.5.2 Entity Relationship Diagram

Perancangan basis data dengan menggunakan model *entity relationship* dengan menggunakan *Entity Relationship Diagram* (ERD)[1]. Terdapat tiga notasi dasar yang bekerja pada model ERD, notasi tersebut dapat dilihat pada Tabel 2.4.

Tabel 2.5 Simbol Entity Relationship Diagram

Simbol	Keterangan
	<p>Entitas</p> <p>Adalah sesuatu apa saja yang ada didalam system, nyata maupun abstrak dimana data yang disimpan atau dimana terdapat data.</p>
	<p>Atribut</p> <p>Adalah sesuatu yang menjelaskan apa yang sebenarnya yang dimaksud entitas maupun relationship, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan relationship.</p>
	<p>Relationship atau relasi</p> <p>Adalah hubungan alamiah yang terjadi karena entitas.</p>
	<p>Garis atau Link</p> <p>Merupakan garis penghubung antara relasi dengan entitas dan entitas dengan atribut.</p>

Terdapat 3 macam kardinalitas relasi, yaitu:

1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

Tingkat satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut.

- a. *One to Many* (satu ke banyak)
- b. *Many to One* (banyak ke satu)

3. *Many to Many*

Tingkat hubungan banyak terjadi jika tiap kejadian pada sebuah entitas kan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari entitas pertama, maupun dilihat dari sisi yang kedua.