BAB 2
LANDASAN TEORI

2.1 Penjadwalan

2.1.1 Definisi Penjadwalan

Penjadwalan berkaitan dengan segala aktivitas untuk membuat sebuah jadwal. Dalam kamus Collins Concise (4^{th} Edition) menyatakan bahwa jadwal adalah susunan tabel kegiatan berdasarkan waktu dan tempat. Kegiatan ini biasanya pertemuan antar beberapa pihak pada waktu dan tempat yang sama. Jadwal harus memenuhi beberapa persyaratan dan memenuhi keinginan semua pihak yang terlibat sebaik mungkin. Waktu dari aktivitas harus disusun sedemikian rupa sehingga tidak ada salah satu komponen pun memliki lebih dari satu kegiatan pada waktu yang sama [**7**].

Penjadwalan kuliah merupakan pengaturan perencanaan belajar mengajar yang meliputi mata kuliah, dosen, waktu dan ketersediaan ruangan. Pada umumnya penjadwalan kegiatan belajar mengajar disajikan dalam sebuah tabel hari dalam seminggu yang terdiri dari slot waktu yang terdiri dari mata pelajaran, hari, jam, serta pengajar yang sesuai dengan mata pelajaran yang diajarkan. **[**8**]**

2.1.2 Batasan dalam Penjadwalan

Dalam masalah penjadwalan memiliki beberapa macam batasan yang dapat menyebabkan output yang dihasilkan menjadi salah. Dalam menerapkan batasan dalam suatu masalah, biasanya tidak terlalu sama untuk setiap masalah **[**8]. Batasan tersebut terdiri dari :

1. *Edge constraint*

Edge constraint adalah batasan yang mengatur dua kejadian tidak boleh menempati satu slot waktu yang sama. Contohnya pada hari Senin jam 07.30 sampai 08.10 tidak mungkin guru A mengajar di ruang kelas 1A dan mengajar di ruang kelas 5A.

2. *Ordering constraint*

Ordering constraint adalah batasan yang menjaga urutan kejadian dalam timetable. Contoh mata pelajaran A harus dilakukan sebelum mata pelajaran B. Biasanya masalah batasan ini jarang digunakan karena dapat menyebabkan penjadwalan menjadi lebih rumit.

3. *Event-spread constraint*

Event-spread constraint adalah batasan yang mengatur penyebaran kejadian pada suatu timetable. Contoh dalam menjalani ujian, seorang siswa diperbolehkan max 3 mata pelajaran yang diujikan dalam 1 hari.

4. *Present specification and exclusion*

Present specification and exclusion adalah menentukan terlebih dahulu slot waktu yang akan digunakan oleh suatu kejadian sebelum proses pencarian solusi dilakukan. Contoh mata pelajaran agama digabung dengan kelas 1-A dan 1-B maka

akan ditentukan waktu yang sama untuk kedua kelas tersebut agar tidak terganggu untuk menyusun mata pelajaran yang lain.

5. *Capacity constraint*

Capacity constraint adalah batasan yang berhubungan dengan kapasitas ruangan. Untuk masing-masing kelas hanya boleh diisi sebanyak 40 siswa.

6. *Hard and soft constraint*

*Hard constrain*t adalah batasan yang sama sekali tidak boleh dilanggar, sedangkan *soft constraint* adalah batasan yang diusahakan semaksimal mungkin tidak dilanggar namun jika dilanggar, hal tersebut masih dapat diterima.

2.2.3 Pembagian penjadwalan akademik

Tiga pembagian dari penjadwalan akademik (*academic timetables*) [8], antara lain :

1. Penjadwalan Sekolah (*School Timetabling*)

Pada penjadwalan sekolah setiap kelas memiliki mata pelajaran tertentu serta memiliki ruangan tertentu dimana proses kegiatan belajar mengajar dilaksanakan. Pada dasarnya isi dari jadwal diatur oleh kurikulum dimana jumlah dari waktu

tiap mata pelajaran yang diajar dalam seminggu sering ditetapkan secara nasional. Setiap kelas terdiri dari seorang pengajar, yang harus ditempati saat pelajar tiba di sekolah hingga meninggalkan sekolah dan memiliki seorang guru tertentu yang akan bertanggung jawab atas kelas tersebut dalam sebuah periode waktu tertentu . Pengajar biasanya dialokasikan di awal proses penjadwalan, yang menjadi masalah adalah menyesuaikan pertemuan dari pengajar dengan kelas untuk slot waktu tertentu sehingga setiap pengajar tertentu mengajar tiap kelas yang diwajibkan kepadanya. Setiap kelas atau pengajar tidak dapat terlibat lebih satu pertemuan pada saat waktu yang bersamaan.

2. Penjadwalan Mata Kuliah (*Course Timetabling*)

Penjadwalan mata kuliah mencakup kumpulan scheduling dari perkuliahan, dimana dalam setiap mata kuliah diberikan sejumlah ruangan dan periode waktu. Karakteristik penjadwalan mata kuliah, antara lain:

a. Setiap mahasiswa dapat memiliki jumlah mata kuliah yang berbeda.
b. Ketersediaan ruangan berperan sangat penting.
c. Jika dua ruangan memiliki mahasiswa yang sama, maka ruangan tidak dapat dijadwalkan pada waktu yang sama.

3. Penjadwalan Ujian (*Exam Timetabling*)

Karakteristik penjadwalan ujian, antara lain:

a. Hanya ada satu ujian untuk tiap objek (mata kuliah).
b. Ada banyak batasan yang berbeda, contohnya pada hari yang sama ada mahasiswa yang memiliki ujian yang sangat banyak dan berurutan waktunya tetapi ada juga yang tidak.
c. Satu ujian dapat memiliki lebih dari satu ruangan.

2.2 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence*) didenfinisikan sebagai kecerdasan yang ditunjukan oleh suatu *entitas* buatan. Sistem seperti ini umumnya dianggap computer. Kecerdasan diciptakan dan dimasukkan kedalam suatu mesin

(computer) agar dapat melakukan perkerjaan seperti yang dapat dilakukan manusia [10].

2.2.1 Definisi Kecerdasan Buatan

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, computer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peranan computer semakin mendominasi kehidupan umat manusia. Computer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki seseorang
tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar computer bisa bertindak seperti dan sebaik manusia, maka computer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu pada *artificial intelegence* (AI) akan mencoba untuk memberikan beberapa metoda untuk membekali computer dengan kedua komponen tersebut agar computer bisa menjadi mesin yang pintar.

Pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang antara lain :

a. Sudut pandang kecerdasan

   Kecerdasan buatan akan membuat mesin menjadi cerdas (mampu berbuat seperti apa yang dilakukan oleh manusia)

b. Sudut pandang penelitian

   Kecerdasan buatan adalah suatu studi bagaimana membuat agar computer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia

c. Sudut pandang bisnis

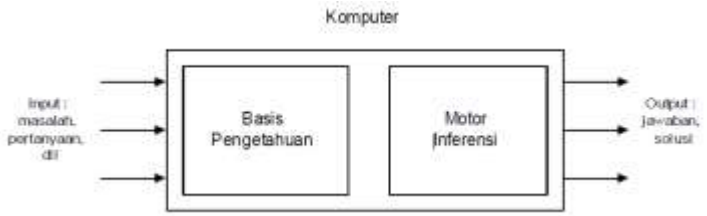
   Kecerdasan buatan adalah kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

d. Sudut pandang pemograman

   Kecerdasan buatan meliputi studi tentang pemograman simbolik, penyelesaian masalah (problem solving) dan pencarian (searching).

   Untuk membuat aplikasi kecerdsan buatan ada 2 bagian utama yang sangat dibutuhkan yaitu :

1. Basis pengetahuan (*knowledge base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor inferensi (*inference engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan Pada Komputer

2.2.2 Kecerdasan Buatan dan Kecerdasan Ilmiah

Kecerdasan alamiah adalah kecerdasan yang dimiliki oleh manusia jika dibandingkan dengan kecerdasan buatan ada beberapa keuntungan kecerdasan buatan disbanding kecerdasan alamiah, yaitu [10] :

1. Lebih permanen

Kecerdasan alamiah akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak diubah

2. Memberikan kemudahan dalam duplikasi dan penyebaran.

Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama dan keahlian itu juga tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

3. Relative lebih murah dari kecerdasan alamiah

Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka watu yang sangat lama.

4. Konsisten dan teliti

Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

5. Dapat didokumentasi

Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.

6. Dapat mnegerjakan beberapa task dengan lebih cepat dan lebih baik dibanding manusia.

Sedangkan keuntunga kecerdasan alamiah dibanding kecerdasan buatan yaitu :

1. Bersifat lebih kreatif

Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.

2. Dapat melakukan proses pembelajaran secara langsung, sementara AI harus mendapatkan masukann berupa symbol dan representasi.

3. Focus yang luas sebagai referensi untuk pengambilan keputusan sebaiknya AI

menggunakan focus yang sempit.

Komputer dapat digunakan untuk mengumpulkan informasi tentang objek, kegiatan (*events*), proses dan dapat memproses sejumlah besar informasi dengan lebih efisien dari yang dapat dikerjakan manusia. Namun disisi lain, manusia dengan menggunakan insting dapat melakukan hal yang sulit diprogram pada komputer, yaitu kemampuan mengenali (*recognize*) hubungan antara hal-hal tersebut, menilai kualitas dan menemukan pola yang menjelaskan hubungan tersebut.

2.3 *Simulated Annealing*

Simulated Annealing adalah salah satu algoritma untuk untuk optimisasi yang bersifat generic. Diperkenalkan oleh Gelett Kirkpatrick dan Vechi(1983) dan Cerny (1985). Algoritma ini berbasiskan probabilitas dan mekanika statistik untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan.

Simulated Annealing dikembangkan berdasarkan ide dari mekanisme perilaku pendinginan dan proses kristalisasi (annealing) material panas. Algoritma ini melakukan peningkatan iteratif untuk memperbaiki solusi yang dihasilkan teknik-teknik penjadwalan heuristik, dalam hal ini adalah sebuah solusi awal yang dibuat dengan teknik heuristic ataupun random, diiterasi secara berulang dengan metode annealing dengan menggunakan perturbasi lokal hingga tidak ada peningkatan lagi atau hingga jumlah iterasi yang diinginkan sudah dicapai [5]. *Simulated Annealing* adalah satu dari algoritma – algoritma terbaik saat ini dalam hampir semua pemecahan masalah optimasi. Dengan sedikit modifikasi, *Simulated Annealing* akan bekerja sangat baik dalam semua permasalahan optimasi [6].

2.4 Algoritma *Simulated Annealing*

Struktur algoritma *Simulated Annealing* secara umum adalah sebagai berikut [12]:

1. Pilih solusi awal *Sol* secara acak.
2. Pembobotan *E=f(Sol).*
3. Bangkitkan solusi baru *Solnew*

4. Hitung Bobot = *f(Solnew).*
5. Jika *f(solnew) > f(Sol)* maka *Sol* = Solnew dan *E = Enew.*
6. Berhenti jika telah mencapai kriteria pemberhentian, jika tidak kembali ke langkah 3.

Pada penelitian ini, langkah ke 2 diambil dari nilai pembobotan yang diperoleh dari pengecekan *constraint-constraint.* Modifikasi penelitian dilakukan pada tahap 3-7, yaitu untuk Solnew merupakan solusi yang memiliki nilai pembobotan yang buruk (langkah 3), sehingga akan dilakukan pembobotan lagi (langkah 4 & 5). Jika masih ada solusi buruk makan dilakukan pengacakan solusi antara solusi baik dan buruk, hal ini dilakukan untuk memperbanyak solusi yang diperoleh.

2.5 Analisis Perancangan Terstruktur

Metode analisis yang digunakan dalam tahapan perancangan sistem dengan menggunakan metode pendekatan sistem berorientasi data atau pendekatan terstruktur. Metode ini diperkenalkan pada tahun 1970, yang merupakan hasil turunan dari pemrograman terstruktur. Metode pengembangan dengan metode terstruktur ini terus diperbaiki sampai akhirnya dapat digunakan dalam dunia nyata [9].

Perancangan ini bertujuan untuk membuat model solusi terhadap problem yang sudah dimodelkan secara lengkap pada tahap analisis terstruktur. Ada empat kegiatan perancangan yang harus dilakukan, yaitu:

a. Perancangan arsitektural, perancangan dengan mengacu pada model analisis yang sesuai (DFD). Langkahnya adalah mengidentifikasi jenis aliran (*transform flow* atau *transaction flow*), kemudian batas-batas aliran (*incoming flow* dan *outgoing flow*), kemudian memetakannya menjadi struktur hirarki modul. Selanjutnya, kita alokasikan fungsi-fungsi yang harus ada pada modul-modul yang tepat.
b. Perancangan data, perancangan struktur data yang dibutuhkan, serta merancang skema basisdata dengan mengacu pada model analisis yang sesuai

(ERD). Perancangan antarmuka dengan pengguna, antarmuka dengan sistem lain, dan antarmuka antar-modul.

c. Perancangan prosedural, perancangan detil dari setiap fungsi pada modul. Notasi yang digunakan bisa berupa flow chart, algoritma, dan lain-lain. Pastikan bahwa model perancangan yang dibuat sudah mengakomodasi kebutuhan non fungsional.

2.5.1 Diagram Konteks

Keadaan sistem secara umum dan hubungan-hubungan sistem tersebut dengan komponen-komponen diluar sistem atau dengan sistem yang lain dapat digambarkan secara logika dengan diagram konteks.

Diagram konteks adalah diagram sederhana yang menggambarkan hubungan antara *entity* luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran yang mewakili keseluruhan sistem. Diagram konteks merupakan DFD level yang paling atas yang hanya terdiri dari satu proses yang menggambarkan sistem atau program secara keseluruhan.

2.5.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah alat yang menggambarkan aliran data melalui sistem dan kerja atau pengolahan yang dilakukan oleh sistem tersebut [9]. Pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun secara komputerisasi.

DFD ini adalah salah satu alat pembuatan yang sering digunakan, khususnya bila fungsi sistem merupakan bagian yang lebih penting dang kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD merupakan perancangan yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Syarat membuat DFD:

a. Pemberian nama untuk setiap komponen DFD
b. Pemberian nomor pada komponen proses

c. Penggambaran DFD serapih mungin agar enak dilihat
d. Penghindaran penggambaran DFD yang rumit
e. Pemastian DFD yang dibentuk itu konsisten secara logika

2.5.3 Kamus data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang input, output dan komponen data store. Pembentukan kamus data didasarkan pada alur data yang terdapat pada DFD. Alur data pada DFD bersifat global (hanya menunjukkan nama alur datanya tanpa menunjukkan struktur dari alur data). Untuk menunjukkan struktur dari alur data secara rinci maka dibentuklah kamus data.

2.6 Database

Semua sistem informasi membuat, membaca, memperbarui, dan menghapus (kadang-kadang disingkat CRUD) data. Data disimpan di dalam *file* dan *database*. *File* adalah sebuah kumpulan *record* yang serupa. Contoh-contohnya mencakup *FILE COSTUMER, FILE ORDER,* dan *FILE PRODUK. Database* adalah kumpulan *file* yang saling terkait. Kata kuncinya adalah "saling terkait". *Database* tidak hanya berupa kumpulan *file*. *Record* pada setiap *file* juga harus memperbolehkan hubungan-hubungan (anggaplah sebagai *"pointer"*) untuk menyimpan *files* lainnya [9].

2.6.1 Database Management System

Database management system adalah perangkat lunak komputer khusus yang disediakan dari vendor-vendor komputer yang digunakan untuk membuat, mengakses, mengontrol, dan mengelola *database*. DBMS sering disebut *database engine.* Mesin ini merepon perintah-perintah khusus untuk membuat struktur *database* kemudian membuat, membaca, memperbarui, dan menghapus *records* pada sebuah *database.* Sistem manajemen *database* dibeli dari sebuah vendor teknologi *database* seperti Oracle, IBM, Microsoft, atau Sybase [9].

Analisis *database*, mendesain struktur data dalam tipe hal *record*, *fields* yang diisikan pada *record* tersebut, dan hubungan yang ada pada tipe-tipe *record*.

Struktur ini didefinisikan untuk sistem manajemen *database* dengan menggunakan bahasa definisi data milik sistem tersebut. *Data Definition Language (DDL)* digunakan oelh DBMS untuk menetapkan secara fisik tipe, *record, field*, dan hubungan struktural. Selain itu DDL menentukan *view database*. *View* tersebut membatasi bagian dari sebuah *database* yang dapat digunakan atau diakses oleh para pengguna dan program yang berbeda.

Sebagian besar sistem menajemen *database* menyimpan baik *user data* maupun metadata – data (atau spesifikasi) mengenai data seperti *record* dan *field*, sinonim-sinonim, hubungan data, aturan-aturan, validasi, *help message*, dan seterusnya.

Sistem manajemen *database* juga menyediakan sebuah bahasa manipulasi data untuk mengakses dan menggunakan data yang disimpan pada aplikasi-aplikasi. *Data Manipulation Language (DML)* digunakan untuk membuat, membaca, memperbarui, dan menghapus *records* pada sebuah database dan untuk menjelajahi diantara *records* dan tipe-tipe *record* yang berbeda. DBMS dan DML menyembunyikan rincian-rincian yang terkait dengan bagaimana *record* diorganisasikan dan dialokasikan pada sebuah *disk.* Pada umumnya DML sangat fleksibel dalam hal ia dapat digunakan untuk membuat, membaca, memperbarui, dan menghapus *records* atau perintah-perintah DML yang dapat "dipanggil" dari sebuah bahasa *host programming* terpisah seperti COBOL, *Visual Basic*, atau *Java.*

2.6.2 Structure Query Language Server

Microsoft SQL Server merupakan produk *Relational Database Management System* (RDBMS) yang dibuat oleh Microsoft. Microsoft SQL juga mendukung bahasa SQL sebagai bahasa untuk memproses *query* ke dalam *database*. Microsoft SQL Server banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi *database* atau penyimpanan data. Berikut ini adalah beberapa fitur yang ada pada SQL Server:

1. XML *Support*. Dengan fitur ini, dapat menyimpan dokumen XML dalam suatu tabel, meng-*query* data ke dalam *format* XML melalui Transact-SQL dan lain sebagainya.

2. *Multi-Instance Support*. Fitur ini memungkinkan dapat dijalankannya beberapa *database engine* SQL Server pada mesin yang sama.
3. *Data Warehousing and Business Intelligence (BI) Improvements*. SQL Server dilengkapi dengan fungsi-fungsi untuk keperluan *Business Intelligence* melalui *Analysis Service.*
4. *Performance and Scalability Improvements.* SQL Server menerapkan *distributed partiotined views* yang memungkinkan untuk membagi *workload* kebeberapa server sekaligus. Peningkatan lainnya juga dicapai di sisi DBCC, *ndexed view,* dan *index reorganitation.*
5. *Query Analyzer Improvements.* Salah satu peningkatan disini adalah T-SQL sudah mendukung UDF (*User-Defieable Function).* Ini memungkinkan untuk menyimpan rutin-rutin kedalam *database engine.*

2.6.3 PHP

PHP merupakan singkatan dari PHP: Hypertext Preprocessor. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan browser.

Secara khusus PHP dirancang untuk membentuk aplikasi web dinamis. Artinya Ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya menampilkan isi database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (Active Server Page), Cold Fusion, ataupun Perl [11].

2.7 Pengujian Sistem

Pada pengujian sistem terdapat rencana pengujian untuk menentukan jenis pengujian yang akan dilakukan.

2.7.1 Pengujian Black Box

Pengujian *black box* merupakan pendekatan komplementer dari teknik *white box*, karena pengujian *black box* diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik *white box*. Pengujian *black box* berfokus

pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program.

Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan [12].

2.7.2 Pengujian Beta

Pengujian Beta (beta testing) didefinisikan sebagai pengujian tahap akhir suatu produk sebelum diluncurkan secara komersial. Pengujian beta biasanya dilakukan terutama pada produk yang berkaitan dengan teknologi informasi. Analogi pengujian beta ini dapat diterapkan untuk produk atau jasa yang lain. Umumnya pengujian tersebut dilakukan dengan mengirimkan atau memberi kesempatan kepada pengguna untuk dapat mencoba produk atau jasa tersebut [12].