

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan Teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Dalam landasan teori ini akan mengulas tentang teori-teori yang menunjang untuk penelitian ini. Landasan teori ini digunakan untuk memperjelas dan mempertajam ruang lingkup atau konstruk variabel yang akan diteliti, meringkas dan menyusun pengetahuan yang ada. Oleh karena itu landasan teori dalam penelitian ini digunakan untuk menjelaskan teori-teori yang berkaitan dengan penelitian ini.

2.2 Buku

Dalam Kamus Besar Bahasa Indonesia disebutkan bahwa buku adalah helai kertas berjilid yang berisikan tulisan untuk dibaca atau kosong untuk ditulis.

Buku menurut Suherman dalam *Fella* Tahun 2014, Secara umum buku dibagi menjadi empat jenis yaitu

- a. Buku Sumber, yaitu buku yang bisa dijadikan rujukan, referensi dan sumber untuk kajian tertentu, biasanya berisi suatu kajian ilmu yang lengkap
- b. Buku Bacaan, adalah buku yang hanya berfungsi untuk bahan bacaan saja, misalnya cerita, legenda, novel dan lain sebagainya.
- c. Buku Pegangan, yaitu buku yang bisa dijadikan pegangan guru atau pengajar dalam melaksanakan proses pengajaran.
- d. Buku teks, yaitu buku yang disusun untuk proses pembelajaran, dan berisi bahan-bahan atau materi pelajaran yang akan diajarkan [4].

2.3 Ekstraksi Informasi

Ekstraksi Informasi adalah Informasi adalah bisa diartikan sebagai sebuah proses untuk menemukan informasi terstruktur dari dokumen tidak terstruktur atau semi terstruktur [5]. Ekstraksi Informasi menghasilkan informasi utama dari sekumpulan berita, menjadi bagian yang paling penting karena dari sinilah bahan beringkasan dihasilkan. Diperlukan analisis khusus agar teknik ekstraksi yang dipakai mampu secara penuh mendukung tugas peringkasan menjadi lebih mudah [6]. Ekstraksi dilakukan berdasarkan analisa teks pengolahan bahasa alami atau Neural Language Processing. Yang dimaksud dengan bahasa alami adalah bahasa normal yang digunakan oleh manusia dalam berkomunikasi, yang biasanya memiliki struktur dan gaya bahasa tertentu. Ekstraksi informasi dilakukan berbasis mesin untuk memahami teks yang berbahasa alami, seperti untuk memahami cerita dan pertanyaan, pembuatan abstrak, dan penerjemahan secara otomatis.

Biasanya ekstraksi informasi dilakukan dalam representasi sistem logika yang bervariasi. Terdapat juga mesin inferensi yang dipakai untuk mengurangi jawaban untuk pertanyaan dan digunakan untuk menyimpulkan jawaban. Pada ekstraksi informasi, biasanya analisa dilakukan secara sintaksis menggunakan basis pengetahuan. Tahapan-tahapan yang ada dalam proses ekstraksi informasi adalah pesan, sintaksis, Semantik, pargmantik, dan maksus yang diinginkan.

2.3.1 Pembentuk Ekstraksi Informasi

Dalam ekstraksi informasi terdapat bagian-bagian pemebtuk,yaitu analisa teks dan strukturisasi data. Berikut adalah penjelasan mengenai analisa teks dan strukturisasi data [7].

1. Analisa Teks

Analisa teks adalah ekstraksi informasi berbasis pada pengolahan bahasa alami untuk mendapatkan arti dari sebuah teks. Dalam memahani bahasa alami, analisa dapat dilakukan berbasis mesin, seperti memahami kisah, pertanyaan, atau menerjemahkan secara otomatis dan merepresentasikan dalam beragam sistem yang logis. Selanjutnya beragam jawaban terhadap berbagai pertanyaan tersebut disimpulkan.

2. Strukturisasi Data

Strukturisasi data biasanya berdasar pada teknik wapping untuk mengatur data yang kurang terstruktur. Menurut definisi Meadows, data adalah pasangan nilai atribut terisolasi yang dapat diolah menjadi informasi. Ekstraksi informasi memberikan pengetahuan dengan tingkat yang lebih dalam dibanding pengambilan informasi yang bisa dilakukan secara tradisional yang menganalisa fakta-fakta dari berbagai dokumen.

2.4 Preprocessing

Preprocessing merupakan tahapan awal dalam mengolah data input sebelum memasuki proses tahapan berikutnya. Preprocessing text dilakukan untuk tujuan penyeragaman dan kemudahan pembacaan [8]. Preprocessing terdiri dari beberapa tahapan yaitu case folding, tokenizing, filtering dan stemming [9].

1. Case Folding

Case folding merupakan tahapan yang mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter.

2. Tokenizing

Tahap tokenizing/pharsing tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Selain itu spasi digunakan untuk memisahkan antar kata.

3. Filtering

Tahap Filtering adalah tahap mengambil kata-kata yang penting dari hasil tokenizing. Proses filtering dapat menggunakan algoritma stoplist atau wordlist. Stoplist/Stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag of word.

4. Stemming

Stemming merupakan suatu proses yang terdapat dalam sistem IR yang menstransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya dengan menggunakan aturan-aturan tertentu.

2.5 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengambilan ciri sebuah objek yang dapat menggambarkan karakteristik dari objek tersebut [10]. Ekstraksi fitur perlu dilakukan dalam ekstraksi informasi, karena data teks yang banyak bentuknya tidak terstruktur tidak memiliki fitur, sehingga tidak bisa langsung dimasukkan kedalam metode rule base, ada tiga jenis fitur yang dapat digunakan untuk ekstraksi informasi, yaitu:

1. Fitur lokal, yaitu ciri-ciri khusus yang ada didalam masing-masing kalimat.
2. Fitur tata letak, yaitu posisi kalimat dalam barisannya masing-masing.
3. Fitur lexicon, yaitu jenis kata kunci yang terkandung dalam kalimat tersebut.

2.6 Rule Base

Rule based adalah sebuah metode dimana aturan yang ada di dalam sistem dibuat sendiri berdasarkan pengetahuan linguistik. Analisis dilakukan pada tingkatan sintaksis dan semantik [11]. Sistem berbasis aturan adalah suatu perangkat lunak yang menjadikan keahlian pakar dalam bentuk aturan-aturan pada suatu domain tertentu untuk menyelesaikan suatu masalah tertentu. Rule base sistem adalah model seherhana yang bisa di adaptasi ke banyak masalah. Namun, jika aturan terlalu banyak, pemeliharaan sistem akan rumit dan terdapat banyak kesalahan dalam kerjanya. Untuk membuat Rule base sistem, ada beberapa hal yang penting harus dimiliki [12]:

1. Sekumpulan fakta yang mewakili working memory. Ini dapat berupa suatu keadaan yang relevan dengan keadaan awal sistem bekerja.
2. Sekumpulan aturan. Aturan ini mencakup setiap tindakan yang harus diambil dalam ruang lingkup permasalahan yang dibutuhkan.
3. Kondisi menentukan bahwa solusi telah ditemukan atau tidak. Hal ini berguna untuk menghindari looping yang tidak akan pernah berakhir.

Rule base dikenal juga dengan sebutan pendekatan hand- crafting atau Knowledge engineering. Disebut pendekatan rule base karena pendekatan ini memanfaatkan keahlian manusia (*human expert*) untuk membuat aturan-aturan (*rules*) secara manual melalui proses pemahaman domain. Human expert atau pakar dituntut untuk bisa memahami SEI dan domain sistem. Jika domain sistem terlalu besar maka dibutuhkan sumber daya manusia yang cukup banyak.

Sistem berbasis aturan (*rule-based sistem*) adalah jalan untuk menyimpan atau memanipulasi pengetahuan untuk menintrepentasikan informasi yang berguna. Biasanya sistem ini diimplementasikan dengan aplikasi Artificial Intelligence. Suatu aturan terdiri dari 2 bagian, yaitu:

1. Antecedent, yaitu bagian yang mengekspresikan situasi atau premis(Pernyataan IF).
2. Consequent, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika situasi atau premis bernilai benar(Pernyataan berawalan THEN).

Umumnya, sebuah aturan dapat mempunyai gabungan beberapa antecedent dengan kata kunci AND (konjungsi), OR (disjungsi), atau kombinasi keduanya.

```
IF <antecedent 1>
AND <antecedent 2>
```

```
.
.
.
```

```
AND <antecedent n>
```

```
THEN <consequent>
```

```
IF <antecedent 1>
```

```
OR <antecedent 2>
```

```
.
```

.
.
OR <antecedent n>
THEN <consequent>

2.7 Informasi Terstruktur

Terstruktur pada kamus bahasa Indonesia adalah sudah dalam keadaan disusun dan diatur rapi. Sedangkan informasi pada kamus bahasa Indonesia adalah pengumpulan atau data yang diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Pengertian Informasi terstruktur ialah suatu kalimat atau teks yang telah mempunyai aturan dan pola yang tersusun rapih.

2.8 OCR (*Optical Character Recognition*)

(Optical Character Recognition) OCR adalah sebuah aplikasi komputer yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk dikonversi ke dalam bentuk file tulisan [13]. Untuk penelitian ini aplikasi OCR digunakan untuk men scan gambar pada image dan dijadikan text. Dengan adanya OCR, Image yang bertulisan tangan, tulisan mesin ketik atau computer text, dapat dimanipulasi.

2.9 Database

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit [14].

Dengan kata lain, semua akses ke basis data akan ditangani oleh DBMS. DBMS ini menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah. Ada beberapa fungsi yang harus ditangani DBMS seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (Database Administrator), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun media penyimpanan (disk) dan juga menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien di sini adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya. Untuk lebih memahami tentang perbedaan basis data dan DBMS, Anda dapat membaca artikel Memahami Basis Data (Database), DBMS dan Program Aplikasi Di MySQL. [15]

2.9.1 Pembagian basis data.

Menurut jenisnya, basis data dapat dibagi menjadi [16]:

a. Basis data flat-file.

Basis data ini ideal untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada dasarnya, basis data flat-file tersusun dari sekumpulan string dalam satu atau lebih file yang dapat diurai untuk mendapatkan informasi yang disimpan. Basis data flat-file cocok untuk menyimpan daftar atau data yang sederhana dan dalam jumlah kecil. Basis data flat-file akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks walaupun dimungkinkan pula untuk itu. Beberapa kendala dalam menggunakan basis data jenis ini adalah rentan pada korupsi data karena tidak adanya penguncian yang melekat ketika data digunakan atau dimodifikasi dan juga adanya duplikasi data yang mungkin sulit dihindari. Salah satu tipe basis data flat-

file adalah file CSV yang menggunakan pemisah koma untuk setiap nilainya.

b. Basis data relasional.

Basis data ini mempunyai struktur yang lebih logis terkait cara penyimpanan. Kata "relasional" berasal dari kenyataan bahwa tabel-tabel yang ada di basis data relasional dihubungkan satu dengan lainnya. Basis data relasional menggunakan sekumpulan tabel dua dimensi yang masing-masing tabel tersusun atas baris (tupel) dan kolom (atribut). Untuk membuat hubungan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu primary key di salah satu tabel dan foreign key di tabel yang lain. Saat ini, basis data relasional menjadi pilihan utama karena keunggulannya. Program aplikasi untuk mengakses basis data relasional menjadi lebih mudah dibuat dan dikembangkan dibandingkan dengan penggunaan basis data flat-file.

2.10 MySQL

MySQL merupakan database server yang bersifat multiuser dan multi threaded. SQL adalah bahasa database standar yang memudahkan penyimpanan, perubahan dan akses informasi. Pada MySQL dikenal dengan istilah database dan tabel. Tabel sendiri merupakan sebuah struktur data dua dimensi yang terdiri dari baris-baris record dan kolom. MySQL termasuk salah satu database open source yang paling banyak digunakan karena selain gratis MySQL pun mempunyai banyak dukungan bahasa pemrograman dan aplikasi sebagai solusi database. [17]

MySQL memiliki keunggulan sebagai berikut:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os Server, dan masih banyak lagi.
2. Open Source. MySQL didistribusikan GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser. MySQL dapat digunakan oleh beberapa pengguna dalam waktu bersamaan tanpa mengalami masalah konflik.
4. Performance Tuning. MySQL mampu memproses lebih banyak query SQL per satuan waktu.
5. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti integer, float, double, char, text, date, datetime, dan lain lain.

MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sistem manajemen database yang bersifat at relational. Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.

2.10.1 SQL (Structured Query Language)

Structured Query Language adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL digunakan untuk mengambil, menambahkan, dan memanipulasi data dari sebuah basis data relasional.

Bahasa SQL terbagi menjadi dua kategori yaitu kategori pertama schema statements kategori pernyataan (statements) untuk membuat objek basis data berupa tabel, index, constraints, dan sebagainya. Kategori kedua (SQL data statements) yaitu pernyataan yang digunakan untuk memanipulasi dan mengambil data yang tersimpan dalam basis data. [18]

2.11 Java

Java merupakan bahasa yang sangat berkembang karena kemampuannya yang dapat mengembangkan banyak hal. Kemampuan pengembangan dengan bahasa pemrograman Java menjadi kemampuan yang paling banyak dibutuhkan di perusahaan teknologi. Java digunakan pada pembuatan aplikasi android mulai dari aplikasinya sampai sistem oprasi android..

Sejarah pemrograman java dimulai ketika pada tahun 1990, Jamis Gosling, pemrogram yang bekerja pada sun Microsystems mendapat tugas menciptakan

program untuk mengendalikan perangkat elektronika rumah tangga. Java adalah penyederhanaan dari C++. Sun sendiri mendeskripsikan java sebagai bahasa pemrograman yang sederhana, berorientasi objek, bertipe statik, tekompilasi, multi-threaded, garbage collected, netral arsitektur, robust, aman, dan dapat diperluas [19]. Pada penelitian ini menggunakan java karna java memiliki kelebihan antara lain sebagai berikut :

1. Sederhana

Java tidak mendukung pointer matematis, casting tipe implisit, struktur atau unions, operator overloading, template, header filer, maupun pewarisan majemuk.

2. Berorientasi objek

Seperti halnya C++, java menggunakan kelas-kelas untuk mengelola kode ke dalam modul-modul logika. Pada saat runtime, program menciptakan objek dari kelas-kelas yang ada.

3. Bertipe statik

Semua objek yang digunakan dalam program harus dideklarasikan sebelum digunakan.

4. Tekompilasi

Sebelum dapat menjalankan program yang ditulis dalam bahasa pemrograman java, program harus dikompilasi oleh kompilator java. Hasil kompilasinya berupa file kode byte, yang mirip dengan file kode mesin yang dapat dieksekusi di bawah sistem operasi apa pun asal memiliki interpreter java.

5. *Multi-threaded*

Program java dapat berisi eksekusi *multi threaded* yang memungkinkan program mengulas citra pada layar pada satu thread, sementara tetap menerima masukan dari keyboard pada thread utamanya.

6. *Garbage collected* (Pembuang sampah)

Program java menyingkirkan sendiri “sampah-sampah” yang tak berguna, artinya program tidak perlu menghapus objek-objek yang dialokasikannya di memori.

7. Bebas arsitektur

Pada dasarnya, java tidak dirancang untuk prosesor ataupun sistem operasi tertentu. Java dirancang untuk bekerja pada berbagai arsitektur prosesor dan berbagai sistem operasi.

8. Tangguh (*robust*)

Karena interpreter java memeriksa seluruh akses sistem yang dilakukan program, maka program java tidak akan membuat sistem menjadi crash. Bila terjadi masalah serius, program java membuat perkecualian (*exception*). *Exception* ini dapat ditangani dan dikelola oleh program tanpa beresiko memacetkan sistem.

9. Aman

Sistem java tidak hanya memverifikasi seluruh akses ke memori tetapi juga menjamin bahwa tidak ada virus yang membongceng pada applet yang sedang berjalan.

10. Dapat diperluas

Program java mendukung metode native yakni fungsi-fungsi yang ditulis dalam bahasa lain, biasanya C++. Dukungan terhadap metode native ini memungkinkan pemrogram menulis fungsi yang bisa dieksekusi secara lebih cepat daripada fungsi yang sama yang ditulis dalam java.

2.12 Object Oriented Programming(OOP)

Booch menyatakan “pemrograman berorientasi objek adalah metode implementasi di mana program diorganisasikan sebagai kumpulan objek yang bekerja sama. Masing-masing objek merepresentasikan instan dari kelas, dan kelas-kelas itu anggota suatu hirarki kelas-kelas yang disatukan lewat keterhubungan pewarisan”. Stroustrup mendeskripsikan pemrograman berorientasi objek sebagai penggabungan abstraksi data dengan pewarisan. Fitur-fitur penting untuk mendukung abstraksi data adalah *constructor* serta *destructor* untuk penciptaan dan pemusnahan objek. Tipe berparameter, mekanisme *exception*, *object coercions* dan

iterator. dalam penelitian ini menggunakan *oop* karena memiliki keunggulan-keunggulan. Keunggulan pendekatan berorientasi objek adalah sebagai berikut:

1. Bekerja yang mendekati kognisi manusia.
2. Menghasilkan sistem yang dibangun di atas bentuk-bentuk antara yang stabil. Dan dengan demikian lebih mampu untuk mengikuti perubahan.
3. Dapat digunakan tidak hanya pada perancangan perangkat lunak tapi juga seluruh proses pengembangan perangkat lunak.
4. Mereduksi resiko pengembangan sistem-sistem kompleks terutama karena pemaduan terjadi menyebar pada siklus kehidupan tidak terjadi dalam sekejap sebagaimana kejadian ledakan besar (*big bang*). [20]

Proses pengembangan aplikasi renovin di bangunan menggunakan OOP dikarenakan OOP menyediakan struktur modular yang jelas untuk program sehingga OOP sangat bagus digunakan untuk mendefinisikan tipe data abstrak di mana detail implementasinya tersembunyi. OOP akan mempermudah dalam memaintain dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada. OOP menyediakan *framework* untuk *library* kode di mana komponen software yang tersedia dapat dengan mudah diadaptasi dan dimodifikasi oleh programmer.

2.13 Unified Modeling Language(UML)

Diagram *use case* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar, dan menjelaskan sistem secara fungsional yang terlihat *user* [21]. Biasanya dibuat pada awal pengembangan. *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem,

mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua feature yang ada pada sistem. Sebuah *use case* dapat memanggil (*include*) fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang memanggil (*include*) dieksekusi secara normal. Sebuah *use case* dapat dipanggil (*include*) oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng-extend *use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Aplikasi renovin menggunakan UML dikarenakan Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Perancangan komponen antarmuka terintegrasi dengan perancangan PL dan struktur data.

2.14 Pengujian

Pengujian adalah suatu proses untuk menemukan error pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian *Software* adalah kegiatan yang ditunjukkan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (*under test*), Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat resiko pada implementasinya [22].

Pengujian perangkat lunak dapat dibedakan menjadi dua yaitu *Black Box Testing* dan *Whitebox Testing*.

- a. *Black Box Testing* atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

- b. *White Box Testing* merupakan metode pengujian perangkat lunak di mana struktur internal diketahui untuk menguji siapa yang akan menguji perangkat lunak. Pengujian ini membutuhkan pengetahuan internal tentang kemampuan sistem dan pemrograman.