



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Judul

##### 2.1.1 Penerapan

Menurut Ali Muhammad (2017:51), “Implementasi (penerapan) merupakan penyediaan sarana untuk melaksanakan sesuatu yang menimbulkan dampak atau akibat terhadap sesuatu”.

Sedangkan menurut Browne dan Wildavsky yang dikutip oleh Firdianti Arinda (2018:19), “Implementasi (Penerapan) adalah perluasan aktivitas yang saling menyesuaikan.”

##### 2.1.2 Pemasaran

Menurut Kotler dan Keller (2015:6) “Pemasaran berhubungan dengan mengidentifikasi dan memenuhi kebutuhan manusia dan masyarakat yang merupakan satu fungsi organisasi dan seperangkat proses untuk menciptakan, mengomunikasikan, dan menyebarkan nilai kepada pelanggan dan mengelola hubungan pelanggan dengan cara yang menguntungkan organisasi dan pemilik sahamnya.”

Sedangkan menurut Manap Abdul (2016:5) “Pemasaran merupakan suatu proses perencanaan dan pelaksanaan konsepsi, penetapan harga, penentuan proses produk, promosi dan tempat atau distribusi, sekaligus merupakan proses sosial dan manajerial untuk mencapai tujuan.”

##### 2.1.3 Sistem Pendukung Keputusan

Menurut Asyriati, dkk. (2018:1) dalam buku Sistem Pendukung Keputusan Teori dan Implementasi. Sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkorelasi satu dengan yang lainnya untuk mencapai tujuan tertentu. Sebagai contoh: sebuah perusahaan memiliki sistem manajerial yang terdiri dari *bottom management*, *middle management*, dan *top manajemen* yang memiliki tujuan untuk mencapai kemajuan masyarakat. Sistem pendukung keputusan dapat



diartikan sebagai suatu sistem yang di rancang yang digunakan untuk mendukung manajemen dalam pengambilan keputusan.

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada tahun 1971 oleh Scoot dalam Turban (2001) dengan istilah *Management Decision System*, kemudian sejumlah perusahaan, lembaga penelitian dan perguruan tinggi mulai melakukan penelitian dan membangun Sistem Pendukung Keputusan, sehingga dari produksi yang dihasilkan dapat disimpulkan bahwa sistem ini merupakan suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur.

#### 2.1.4 *Simple Additive Weighting (SAW)*

Menurut Asyriati, dkk. (2018:21) “Metode *Simple Additive Weighting* merupakan pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan dengan mencari rating kinerja (skala prioritas) pada setiap alternatif di semua atribut.”

Adapun algoritma penyelesaian metode ini adalah sebagai berikut :

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan di jadikan sebagai tolak ukur penyelesaian masalah.
2. Langkah 2 : Menormalisasi setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai rating kinerja.
3. Langkah 3 : Menghitung nilai bobot prefensi pada setiap alternatif.
4. Langkah 4 : Melakukan perangkaian.

Berikut rumus yang digunakan pada metode *Simple Additive Weighting* yaitu :

1. Menormalisasikan setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai rating kinerja

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$



## 2. Menghitung nilai bobot prefensi pada setiap alternatif

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Keterangan :

$V_i$  = Nilai Bobot Preferensi dari setiap alternative

$W_j$  = Nilai Bobot Kinerja

$R_{ij}$  = Nilai Rating Kinerja

### 2.1.5 Penerapan Metode *Simple Additive Weighting* (SAW) dalam Sistem Pendukung Keputusan Menentukan Target Lokasi Pemasaran Produk Pemurni Air pada PT Luxindo Raya

Aplikasi Penerapan Metode Simple Additive Weighting (SAW) dalam Sistem Pendukung Keputusan Menentukan Target Lokasi Pemasaran Produk Pemurni Air pada PT Luxindo Raya merupakan aplikasi yang bertujuan untuk menjadi sarana penunjang bagi pengambil keputusan dalam menentukan target lokasi pemasaran produk pemurni air. Aplikasi ini menampilkan hasil perhitungan kedalam list yang berurutan berdasarkan ranking setiap daerah.

## 2.2 Teori Khusus

### 2.2.1 *Unified Modelling Language* (UML)

Menurut Sukanto dan Shalahuddin (2018:13), *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

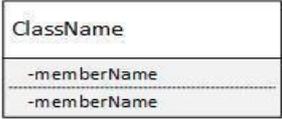
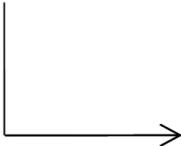


### 2.2.2 Class Diagram

Menurut Sukamto dan Shalahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

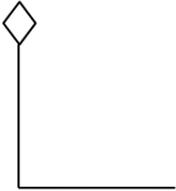
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**Tabel 2.1** Simbol Class Diagram

No	Simbol	Deskripsi
1	Kelas  	Kelas pada struktur sistem
2	Antarmuka/ <i>interface</i>  	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	Asosiasi/ <i>association</i>  	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	Asosiasi berarah/ <i>directed association</i>  	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .



Lanjutan Tabel 2.1

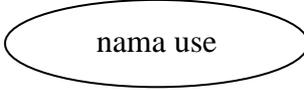
No	Simbol	Deskripsi
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)
6	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

Sumber : Sukamto dan Shalahuddin (2018:141)

### 2.2.3 Use Case Diagram

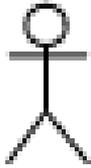
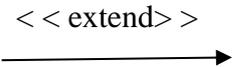
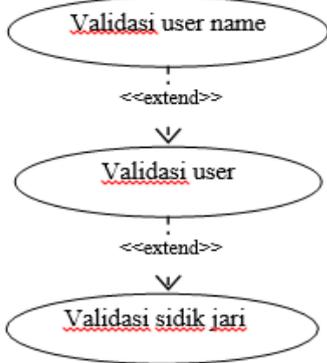
Menurut Sukamto dan Shalahuddin (2018:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
1	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anta unit atau aktor; biasanya dinyatakan menggunakan kata kerja di awal.

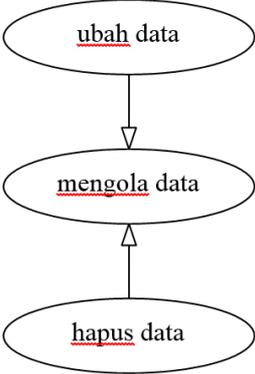
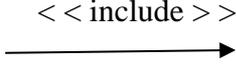


Lanjutan Tabel 2.2

No	Simbol	Deskripsi
2	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4	Ekstensi / <i>extend</i> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal :</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, yang <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>



Lanjutan Tabel 2.2

No	Simbol	Deskripsi
5	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
6	Menggunakan / <i>include</i> / <i>uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :

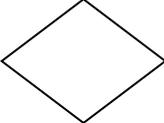
Sumber : Sukamto dan Shalahuddin (2018:155)



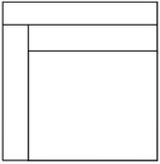
### 2.2.4 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel 2.3** Simbol Activity Diagram

No	Simbol	Deskripsi
1	Start / status awal ( <i>Initial State</i> ) 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	End / status akhir ( <i>final state</i> ) 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan **Tabel 2.3**

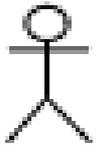
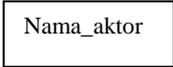
No	Simbol	Deskripsi
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

**Sumber :** Sukamto dan Shalahuddin (2018:161)

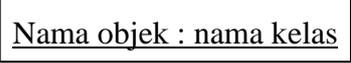
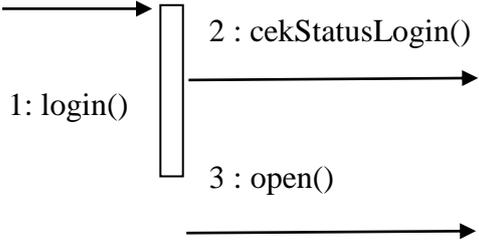
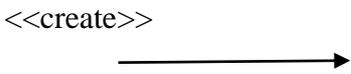
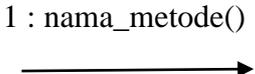
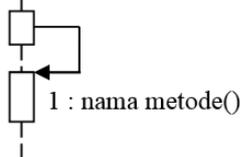
### 2.2.5 Sequence Diagram

Menurut Sukamto dan Shalahuddin (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

**Tabel 2.4** Simbol Sequence Diagram

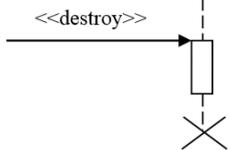
No	Simbol	Deskripsi
1	Aktor  atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
2	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.

Lanjutan **Tabel 2.4**

No	Simbol	Deskripsi
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 



Lanjutan Tabel 2.4

No	Simbol	Deskripsi
		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Sukamto dan Shalahuddin (2018:165)

## 2.3 Teori Program

### 2.3.1 Pengertian *Flutter*

Menurut Raharjo Budi (2019:1) dalam buku Pemrograman Android dengan Flutter, Flutter adalah software development kit (SDK) buatan google yang berfungsi untuk membuat aplikasi mobile menggunakan bahasa pemrograman Dart, baik untuk Android maupun iOS. Dengan flutter, aplikasi

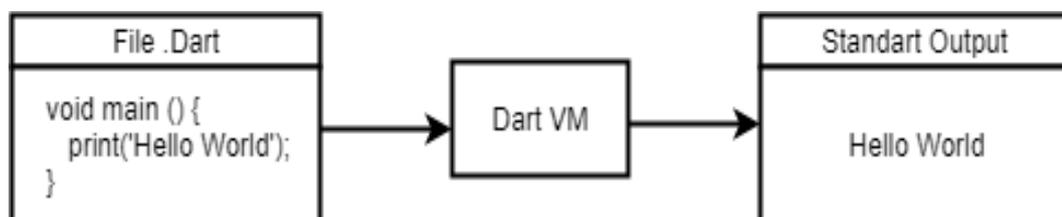


Android dan iOS dapat dibuat menggunakan basis kode dan Bahasa pemrograman yang sama, yaitu Dart, Bahasa pemrograman yang juga diproduksi oleh Google pada tahun 2011. Sebelumnya, aplikasi murni (native) untuk Android perlu dibuat menggunakan Bahasa pemrograman Java atau Kotlin. Sedangkan aplikasi iOS perlu dibuat menggunakan Bahasa pemrograman Objective-C atau Swift. Flutter ditujukan untuk mempermudah dan mempercepat proses pengembangan aplikasi *mobile* yang dapat berjalan di atas Android dan iOS, tanpa harus mempelajari dua Bahasa pemrograman secara terpisah.

### 2.3.2 Pengertian Dart

Menurut Raharjo Budi (2019:1) dalam buku Pemrograman Android dengan Flutter, Dart adalah bahasa pemrograman yang diproduksi oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Dart pertama kali dikenalkan pada 10 Oktober 2011.

Dart dapat digunakan untuk membuat aplikasi server (berbentuk command-line interface), web, maupun mobile (*Android dan iOS*). Aplikasi Dart dieksekusi secara langsung melalui *Dart Virtual Machine* (VM) tanpa melalui proses penerjemahan ke kode objek (bytecode) terlebih dahulu. Gambar berikut menunjukkan proses eksekusi aplikasi Dart.



**Gambar 2.1** Proses Eksekusi Aplikasi Dart

### 2.3.3 Hypertext Processor (PHP)

Menurut Yudhanto dan Prasetyo (2019:9) “PHP atau *Hypertext Processor* adalah Bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web.”



Sedangkan menurut Sulistiono Heru (2019:5) “PHP (*Hypertext Processor*) adalah Bahasa pemrograman yang digunakan untuk membuat *website* atau situs dinamis dan menangani rangkaian Bahasa pemrograman antara *client side scripting* dan *server side scripting*.”

Dari pengertian-pengertian diatas, dapat disimpulkan PHP atau *Hypertext Processor* adalah Bahasa pemrograman yang digunakan untuk pemrograman *script server side*.

#### 2.3.4 Mysql

Menurut Sumarlinda (2015:20) “MySQL adalah *multiuser database* yang menggunakan bahasa *structured query language* (SQL), MySQL dalam operasi *client-server* melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*.

Menurut Sidik (2017:301) “MySQL merupakan *software database* yang termasuk paling populer dilingkungan *Linux*, kepopuleran ini karena ditunjang karena performasi *query* dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah. MySQL telah tersedia juga dilingkungan *Windows*”.

Berdasarkan pengertian diatas, bahwa dapat disimpulkan MySQL adalah sebuah database yang digunakan untuk menyimpan data dalam tabel terpisah, mysql dapat berjalan stabil pada berbagai sistem operasi seperti windows,linux dan lainnya serta dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan.

#### 2.4 Penelitian Terdahulu

Menurut penelitian oleh Arifin dan Fadillah (2016) telah ditemukan sebuah sistem informasi dengan kombinasi metode *Weighted Product* (WP) dan metode *Simple Additive Weighting* (SAW) dalam penyeleksian kawasan untuk penanaman bakau. Metode ini dipilih karena merupakan metode yang sederhana secara perhitungan (komputasinya sederhana). Metode WP memiliki keunggulan dalam membandingkan satu alternatif dengan alternatif lainya dengan mengalikan



sejumlah rasio satu untuk setiap kriteria. Metode SAW memiliki keunggulan dalam melakukan transformasi linier terhadap data mentah.

Yanto (2017) melakukan sebuah penelitian yang menyebutkan, metode *Simple Additive Weighting* (SAW) pada proses investasi pendidikan pada perguruan tinggi diharapkan dapat membantu pihak manajemen dalam pengambilan keputusan untuk melakukan investasi yang terbaik berdasarkan alternatif yang dipilih demi tercapainya perguruan tinggi yang berkualitas sesuai dengan visi misi dan tujuan perguruan tinggi.

Dalam penelitian Zubaeti (2017) mengatakan bahwa metode *Simple Additive Weighting* (SAW) digunakan untuk menentukan kelayakan tempat praktik kerja industri sistem ini dilengkapi dengan fitur-fitur untuk mengelola kegiatan administrasi industri pelatihan kerja, untuk merekap jurnal harian, dan untuk merekap laporan pengawasan.

Fauzan, dkk. (2017) melakukan penelitian yang menyatakan penentuan Penerima Beasiswa Bidik Misi ditentukan berdasarkan kriteria tertentu. Sistem pendukung keputusan merupakan sebuah sistem yang digunakan untuk menentukan alternatif penerima Beasiswa Bidik Misi di Politeknik Negeri Banjarmasin yang paling mendekati kriteria yang telah ditentukan dengan menggunakan metode *Simple Additive Weighting* (SAW). Konsep dasar dari metode *Simple Additive Weighting* ini adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Pengujian dilakukan dengan data baru (data uji) dari alternatif penerima Beasiswa Bidik Misi data uji tersebut dinormalisasi sesuai dengan jenis atribut kriteria (*benefit* atau *cost*).

Dalam penelitian Hafni, dkk. (2017) mengatakan Peningkatan gaji sangat mempengaruhi motivasi dan produktivitas karyawan dalam melaksanakan dan menyelesaikan pekerjaan. Untuk menentukan besarnya kenaikan gaji. Pemanfaatan sistem pendukung keputusan dengan menggunakan metode *Simple Additive Weighting* (SAW) membantu manajer untuk membuat pengambilan keputusan yang lebih cepat dan lebih akurat.