

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang dilakukan oleh peneliti mencari beberapa penelitian yang terkait, baik itu secara objek penelitian atau pun secara metode. Peneliti mengambil artikel dari beberapa sumber yang didapat. Perbedaan dengan penelitian yang akan dilakukan. Penulis telah melakukan tinjauan pustaka pada hasil penelitian terdahulu, dan ditemukan beberapa penelitian tentang sistem pemesanan dan informasi *wedding organizer* berbasis web yang menggunakan teknologi *framework codeigniter*. Berikut adalah deskripsi dan tabel mengenai sistem pemesanan dan informasi *wedding organizer* berbasis web.

Dari penelitian Sukindro Setyo Prabowo mahasiswa STMIK Akakom Yogyakarta (Prabowo, 2019) berjudul “Sistem Reservasi *Wedding Organizer* berbasis web menggunakan *Framework Codeigniter* dan *Bootstrap* yang dilakukan di *Romantic Wedding*”. Pembuatan sistem menggunakan teknologi *Framework Codeigniter* dan *Bootstrap* sehingga tampilan sistem dapat menyesuaikan dengan perangkat yang digunakan untuk mengaksesnya. Serta dalam sistem ini ditambahkan fitur undangan pernikahan digital yang diharapkan dapat menambah daya tarik bagi perusahaan.

Dari penelitian Agung Setiajati mahasiswa Jurusan Teknik

Informatika Fakultas Teknik Universitas Muhammadiyah Malang 2016 (Setiajati , 2016) berjudul “Implementasi Sistem Informasi Jasa *Wedding Organizer* dengan Fitur Pemilihan Paket Otomatis Menggunakan Algoritma Greedy, dalam sistem ini menggunakan Algoritma Greedy” yang memberikan rekomendasi paket sesuai dengan biaya dan jumlah undangan yang diinginkan. Algoritma greedy membentuk solusi langkah perlangkah (*step by step*). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi, karenanya pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah selanjutnya. Sebagai contoh, jika kita menggunakan algoritma greedy untuk menempatkan komponen pada papan sirkuit. Sekali komponen telah diletakan dan dipasang, maka tidak dapat dipindahkan lagi. Pada setiap langkah diperoleh optimum lokal. Bila algoritma berakhir, kita berharap optimum lokal menjadi optimum global.

Dari penelitian Fahrudin Prasetyo mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga (Prasetyo , 2017) berjudul “Perancangan Sistem Informasi Pemesanan *Event Organizer* Berbasis *Website* Menggunakan *Framework CodeIgniter*”. Pemesanan *event organizer* berbasis *website* yang diharapkan dapat mempermudah Munji EO dalam pengelolaan data pemesanan maupun keuangan. Dengan sistem informasi yang akan dibangun diharapkan dapat memberikan

informasi kepada pihak Munji EO seperti paket acara yang sering dipesan, atau pelanggan yang sering menggunakan jasa Munji EO, sehingga dapat dijadikan sebagai acuan pengambilan keputusan untuk mengembangkan usaha Munji EO. Bagi pelanggan, sistem informasi mempermudah dalam melakukan pemesanan jasa Munji EO karena dapat dilakukan secara online dimanapun tanpa harus datang ke kantor pemasaran Munji EO. Sistem informasi pemesanan *event organizer* berbasis web dibangun dengan menggunakan *framework CodeIgniter* karena *framework CodeIgniter* mendukung *Model View Controller (MVC)* sehingga pembuatan sebuah *website* akan menjadi lebih terstruktur. *Framework CodeIgniter* juga menyediakan bermacam-macam *library* yang akan mempermudah pembangunan *website* diantaranya adalah *library session*, dan *library dompdf* yang akan digunakan dalam penelitian ini. Kerangka tampilan dibangun dengan menggunakan *framework Bootstrap* untuk menghasilkan tampilan yang responsif.

Dari hasil penjabaran penelitian terkait dapat dilihat tabel perbandingan penelitian yang dapat dilihat dalam Tabel 2.1.

Tabel 2. 1 Kajian Penelitian Terdahulu

No	Judul	Penulis (Tahun)	Metode / Teknologi	Hasil
1.	Sistem Reservasi Wedding Organizer Berbasis Web Menggunakan	Sukindro Setyo Prabowo, (2019)	Codeigniter & Bootstrap	Sistem dibangun dengan <i>responsive design</i> dan dapat mempermudah proses reservasi dan menjadi

No	Judul	Penulis (Tahun)	Metode / Teknologi	Hasil
	<i>Framework Codeigniter & Bootstrap</i> (Studi Kasus : Romantic Wedding)			media promosi baru
2.	Implementasi Sistem Informasi Jasa <i>Wedding Organizer</i> dengan Fitur Pemilihan Paket Otomatis Menggunakan Algoritma Greedy	Agung Setiajati (2016)	PHP dengan Algoritma Greedy	Sistem ini menghasilkan memberikan rekomendasi paket sesuai dengan biaya dan jumlah undangan yang diinginkan
3.	Perancangan Sistem Informasi Pemesanan <i>Event Organizer</i> Berbasis <i>Website</i> Menggunakan <i>Framework CodeIgniter</i>	Fahrudin Prasetyo (2017)	PHP <i>Framework CodeIgniter</i>	sistem informasi mempermudah dalam melakukan pemesanan jasa Munji EO karena dapat dilakukan secara online dimanapun tanpa harus datang ke kantor pemasaran Munji EO
4.	Implementasi Sistem Informasi Pemesanan Jasa <i>Wedding Organizer</i> Menggunakan Teknologi <i>Framework Codeigniter</i>	Diusulkan Aldi Pradeva Hermawan (2024)	PHP <i>Framework CodeIgniter</i>	Sistem yang akan dibangun menjadikan layanan pemesanan paket yang dapat dipilih langsung oleh pengguna dengan design responsive design dan dilengkapi dengan pembayaran dengan <i>payment gateway</i>

Dari hasil kajian pustaka yang dilakukan penelitian saat ini, yaitu tentang Implementasi Sistem Informasi Pemesanan Jasa *Wedding Organizer* Menggunakan Teknologi *Framework Codeigniter*. Pembuatan sistem menggunakan teknologi *Framework Codeigniter* sehingga tampilan akan menjadi lebih dinamis untuk digunakan. Serta dalam sistem ini ditambahkan *payment Gateway* dalam proses

pembayaran dengan *library Midtrans*.

2.2 Dasar Teori

2.2.1 Code Igniter

Codeigniter adalah sebuah web *application framework* yang bersifat *open source* yang digunakan untuk membangun aplikasi PHP dinamis. Tujuan utama pengembangan *codeigniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. (<https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>)

Codeigniter menyediakan berbagai macam *library* yang dapat mempermudah pengembangan. *Codeigniter* dibangun menggunakan konsep Model-View-Controiller (*MVC*) *development pattern*. Dalam *Codeigniter*, *browser* berinteraksi melalui *controller*. *Controller* yang akan menerima dan membalas semua *request* dari *browser*. Ketika *controller* membutuhkan data maka *controller* akan meminta ke model. Sedangkan untuk tampilan ke *user* akan ditangani oleh *view*. Jadi, otak dari aplikasi ada di *controller*, muka aplikasi ada di *view* dan data berada di model. Ada beberapa kelebihan *codeigniter* dibandingkan dengan kerangka kerja PHP lain:

- a. Performa sangat cepat : salah satu alasan tidak menggunakan kerangka kerja karena eksekusinya yang lebih lambat daripada

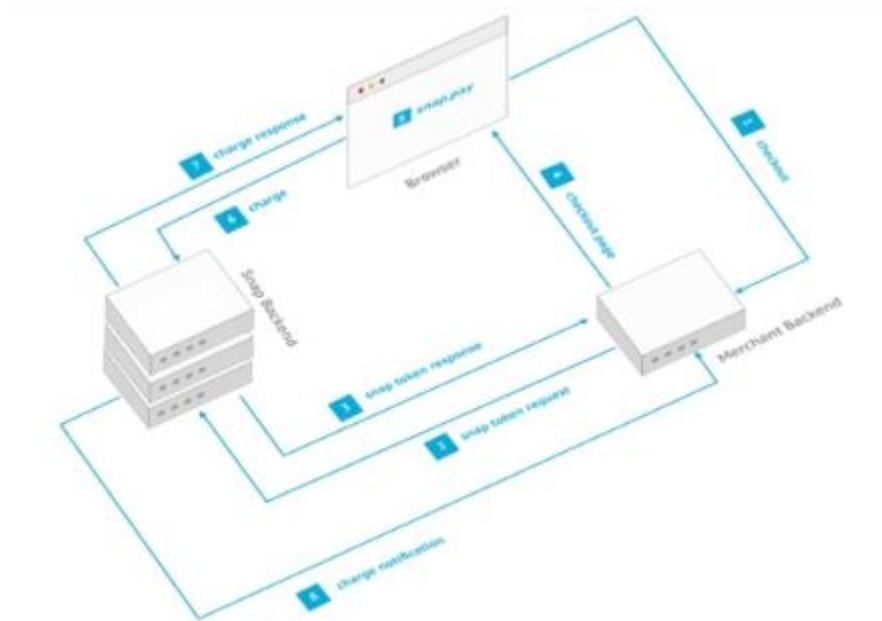
PHP *from the scratch*, tetapi *Codeigniter* sangat cepat bahkan mungkin bisa dibilang *codeigniter* merupakan kerangka kerja yang paling cepat dibanding kerangka kerja yang lain.

- b. Konfigurasi yang sangat minim (*nearly zero configuration*): tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database*, PHP atau autoload PHP, tetapi untuk menggunakan *codeigniter* dengan *setting standard* anda hanya perlu mengubah sedikit saja file pada *folder config*.

Banyak komunitas: dengan banyaknya komunitas CI ini memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

2.2.2 Midtrans Payment Gateway

Midtrans merupakan salah satu *platform* untuk melakukan *payment gateway*. *Midtrans* merupakan *system* pembayaran yang dapat memfasilitasi penjual dan pembeli untuk melakukan transaksi. *Midtrans* menyediakan *tools* terintegrasi ke *e-commerce* sesuai kebutuhan pembayaran secara *online* dengan kartu debit, kartu kredit, bahkan penarikan uang, dan pengiriman uang. Dengan *midtrans* dapat melakukan pembayaran belanja *online*, donasi, produk berlangganan, dan penarikan uang dengan mudah, cepat dan aman.



Gambar 2.1 Alur Kerja *Midtrans*

Cara kerja *midtrans*, yakni pelanggan melakukan *checkout* pada halaman *website*. Selanjutnya server *system* pemesanan akan membuat permintaan API ke *backend snap* untuk mendapatkan *snap* token. *Snap backend* milik *midtrans* merespon panggilan API dengan *snap* token. Server *system* pemesanan akan membangun halaman HTML dan mengirimkannya kembali ke *browser* guna menampilkan *pop up* pembayaran. Kemudian pelanggan memilih salah satu metode pembayaran dan mengklik tombol bayar untuk melakukan pembayaran. Pengguna kemudian mengklik tombol konfirmasi. Selanjutnya sistem mengirimkan detail pembayaran ke *backend snap* kemudian memproses detail dan merespons dengan status pembayaran. Jika pembayaran telah tersimpan, maka *snap backend*

memberitahu *server system* pemesanan tentang status pembayaran. Apabila terdapat pembayaran dari pelanggan yang masuk ke *midtrans*, *snap backend* akan memberitahu *server system* pemesanan dan merubah status pemesanan.

2.2.3 *Wedding Organizer (WO)*

Wedding organizer merupakan jasa perencana pesta pernikahan yang membantu calon pengantin dalam hal perencanaan dan pengelolaan pesta pernikahan. Ketika Anda menggunakan jasa *wedding organizer*, mereka akan membantu memenuhi kebutuhan Anda dan mengatasi masalah yang mungkin terjadi dalam mempersiapkan pesta pernikahan.

WO akan menawarkan berbagai *service* yang dimiliki. Mulai dari *wedding planner*, yaitu merencanakan seluruh konsep pernikahan dari awal atau paket pada hari H saja. Sesuaikan *service* yang dimiliki sesuai dengan *budget* yang sudah disepakati bersama.

Saat bertemu dengan tim WO pastikan sudah tahu berapa jumlah undangan. Jumlah tamu undangan ini sangat berpengaruh ke *service* dari WO. Biasanya WO akan menentukan harga sesuai dengan jumlah tamu undangan.

Untuk merencanakan persiapan pernikahan harus sudah punya tanggal pasti pernikahan kapan akan digelarnya pernikahan. Ini krusial, dikarenakan berpengaruh dengan *available date* dari WO apakah bisa bekerjasama di hari itu atau tidak.

Mempunyai bayangan konsep pernikahan. Walaupun masih bayangan kasar namun setidaknya bisa menyampaikan kepada pihak WO dan pada kesempatan ini kita bisa melihat responnya apakah meyakinkan dan terpercaya untuk berkerjasama atau tidak.

2.2.4 UML

UML (*Unified Modeling Language*) adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi. UML menyediakan model-model yang tepat, tidak ambigu, dan lengkap. Secara khusus UML menspesifikasi langkah-langkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak (Nugroho, Hardiyanto, & Rohimi, 2020)

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi *object* dan *design* berorientasi object (OOAD&D/object oriented analysis and design) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan). Tujuan UML diantaranya adalah :

- a. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

2.2.5 Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

Tabel 2.2 Simbol-simbol Use Case Diagram

Simbol	Keterangan
 AKTOR	AKTOR Menggambarkan segala pengguna <i>software</i> aplikasi (sistem). Aktor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan oleh sistem.
	USE CASE Menjelaskan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu
	ASOSIASI Komunikasi antara <i>actor</i> dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan <i>actor</i> .
	DEPENDENCY Relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain
	GENERALISASI

Simbol	Keterangan
	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. <i>INCLUDE</i>
<<include>>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.
	

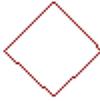
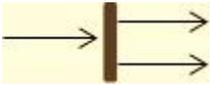
Activity Diagram

Activity mendefinisikan bahwa, diagram memodelkan proses bisnis dan urutan aktifitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *work flow* dari suatu aktifitas lainnya atau dari aktifitas ke status.

Activity Diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem.

Tabel 2. 3 Simbol-simbol diagram Aktifitas

Simbol	Keterangan
	STATUS AWAL Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	AKTIVITAS Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

Simbol	Keterangan
	PERCABANGAN Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	PENGGABUNGAN Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	STATUS AKHIR Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	SWIMLINE Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

***Class* Diagram**

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class* diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class* diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class diagram memiliki 3 area pokok (utama) yaitu nama, atribut dan operasi. Nama berfungsi untuk member identitas pada sebuah class, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek didalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. Dalam

mendefinisikan metode yang ada didalam class diagram harus diperhatikan yang namanya *cohesion* dan *coupling*. *Cohesion* adalah ukuran keterkaitan sebuah instruksi disebuah metode, *coupling* adalah ukuran keterkaitan antar metode. Didalam *class* diagram terdapat hubungan antar *class* secara konseptual, yang disebut relasi antar *class*.

Tabel 2.4 Simbol-simbol diagram *Class*

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.5 Basis Data

Menurut Connolly dan Begg (2010), basis data adalah sebuah

kumpulan data yang secara logis terkait dan dirancang untuk memenuhi suatu kebutuhan informasi dari sebuah organisasi. Menurut Indrajani (2015), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

Seperti yang telah di jelaskan diatas, basis data merupakan sekumpulan file-file yang saling berhubungan dan hubungan tersebut biasanya dapat ditunjukkan dengan kunci dari setiap file yang ada. Satu basis data menunjukkan satu kumpulan data yang digunakan dalam suatu lingkup instansi atau perusahaan.

Berikut ini kegunaan basis data dalam mengatasi masalah penyusunan data :

- a. Redundansi dan inkosisten data.
- b. *Multiple user*.
- c. Kesultan mengakses data.
- d. *Security* (keamanan).
- e. Kebebasan data.

2.2.6 PHP

Bahasa pemrograman berbasis web yang memiliki file dengan extensi file.php dan tidak dapat diakses tanpa adanya web server adalah PHP. PHP *Hypertext Preprocessor* adalah suatu bahasa scripting khususnya digunakan untuk *web development*. PHP juga

merupakan bahasa pemrograman berbasis web yang dibuat secara khusus untuk membangun aplikasi berbasis web. PHP (*hypertext preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML (Madcoms, 2016).

Dari beberapa definisi diatas penulis menyimpulkan bahwa PHP adalah bahasa pemrograman berbasis web yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada HTML. PHP memiliki sifat server *side scripting* sehingga untuk menjalankan PHP harus menggunakan web server.

2.2.7 MySQL

MySQL adalah jenis database server yang sangat populer. MySQL termasuk jenis RDBMS (*Relational Database Management System*), itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL (Wahana, 2014).

Mysql merupakan sebuah *database* server yang free artinya kita bebas menggunakan database ini Mysql adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structure Query Language*). Mysql merupakan sebuah *database* server yang *free* artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa membeli atau membayar lisensinya.

MySQL merupakan hasil ciptaan dari Michael Widenius, David Axmark, dan Allan Larson. Pada tahun 1995 mereka lalu

mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal diciptakannya MySQL yaitu untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL AB.

Database MySQL merupakan perangkat lunak database yang berbentuk *database* relasional atau dalam bahasa basis data sering disebut dengan *Relation Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan bernama SQL. Perbedaan antara MySQL dan SQL, MySQL adalah program *database* server sedangkan SQL adalah bahasa yang digunakan di dalamnya.

2.2.8 Pengujian System

Pengujian *system* merupakan proses mengeksekusi *system* perangkat lunak untuk menentukan apakah *system* perangkat lunak tersebut cocok dengan spesifikasi *system* dan berjalan sesuai dengan lingkungan yang diinginkan. Pengujian *system* sering diasosiasikan dengan pencarian *bug*, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi *system* perangkat lunak (Vikasari, 2018). Pada umumnya ada dua metode untuk melakukan pengujian *system*

1. *Black Box Testing*

Terfokus pada apakah unit program memenuhi kebutuhan yang disebut dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modal, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis

yang diinginkan. Jika unit yang tidak sesuai *outputnya* maka untuk menyelesaikannya akan diteruskan pada pengujian yang kedua, yaitu *white box testing*.

2. *White Box Testing*

White Box Testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki.