

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian terdahulu mengenai sistem informasi berbasis web digunakan sebagai referensi dalam pengembangan sistem Informasi Penjualan *Online* Berbasis Web untuk Central Palantea Hidroponik. Beberapa penelitian terdahulu memberikan landasan yang relevan, baik dari segi metode pengembangan, implementasi teknologi, maupun pemecahan masalah yang dihadapi.

Penelitian oleh Lestari dkk., (2020) membahas pengembangan teknologi hidroponik yang bertujuan memaksimalkan pemanfaatan lahan terbatas di kompleks perumahan. Penelitian ini memberikan wawasan mengenai pengelolaan sumber daya secara efisien di ruang yang terbatas. Metodologi yang digunakan dalam penelitian ini adalah kualitatif. Masalah yang diangkat adalah pemanfaatan lahan terbatas untuk pertanian. Tujuan dari penelitian ini adalah meningkatkan efisiensi operasional pertanian hidroponik di lahan terbatas. Hasil penelitian menunjukkan bahwa teknologi hidroponik dapat secara signifikan meningkatkan produktivitas pertanian di area yang terbatas, serta mengoptimalkan penggunaan sumber daya yang ada.

Penelitian oleh Mutiara dkk., (2023) berfokus pada pengembangan sistem berbasis web untuk mendukung pemasaran produk hidroponik UMKM. Penelitian ini menunjukkan pentingnya integrasi teknologi digital dalam memperluas cakupan pasar dan meningkatkan pemasaran produk. Metodologi yang digunakan dalam penelitian ini adalah *SDLC*. Masalah yang diangkat adalah pemasaran produk hidroponik yang terbatas pada lingkup lokal. Tujuan dari penelitian ini adalah mengembangkan *platform digital* untuk meningkatkan pemasaran produk UMKM. Hasil penelitian menunjukkan bahwa sistem *e-marketing* yang dikembangkan efektif dalam memperluas pemasaran produk.

Penelitian oleh Fitriani & Supriyanto, (2022) bertujuan untuk meningkatkan efisiensi pengelolaan stok dan penjualan. Penelitian ini relevan karena memberikan solusi digital untuk efisiensi operasional perusahaan. Metodologi yang digunakan dalam penelitian ini adalah *Waterfall*. Masalah yang diangkat adalah pengelolaan

penjualan dan stok yang saat ini digunakan tidak efisien. Tujuan dari penelitian ini adalah mengembangkan sistem berbasis *web* untuk meningkatkan efisiensi operasional penjualan dan stok. Hasil penelitian menunjukkan bahwa sistem informasi penjualan yang dikembangkan mendukung efisiensi pengelolaan stok dan pencatatan transaksi.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Penulis	Metodologi	Masalah	Tujuan	Hasil
Lestari dkk. (2020)	Kualitatif	Pemanfaatan lahan terbatas untuk pertanian.	Meningkatkan efisiensi operasional pertanian hidroponik di lahan terbatas.	Sistem pertanian hidroponik untuk ruang sempit yang meningkatkan produktivitas.
Mutiara dkk. (2023)	<i>SDLC</i>	Pemasaran produk hidroponik yang terbatas pada lingkup lokal.	Mengembangkan <i>platform digital</i> untuk meningkatkan pemasaran produk UMKM.	Sistem <i>e-marketing</i> yang efektif dalam memperluas pemasaran produk.
Fitriani & Supriyanto (2022)	<i>Waterfall</i>	Pengelolaan penjualan dan stok yang digunakan tidak efisien.	Mengembangkan sistem berbasis web untuk meningkatkan efisiensi operasional penjualan dan stok.	Sistem informasi penjualan yang mendukung efisiensi pengelolaan stok dan pencatatan transaksi.

Penelitian ini menggunakan metodologi RSSD dan *Design Thinking* untuk mengatasi berbagai kendala operasional di Central Palantea Hidroponik. Beberapa kendala yang dihadapi antara lain sistem pemesanan yang saat ini digunakan menyebabkan keterlambatan transaksi, kurangnya pemantauan stok yang mengakibatkan ketidakseimbangan antara ketersediaan produk dan permintaan pasar, serta pencatatan transaksi yang rentan terhadap kesalahan. Selain itu, terbatasnya jangkauan pemasaran digital serta ketidakefisienan dalam pengelolaan kunjungan yang menyebabkan bentrokan jadwal juga menjadi tantangan utama. Penelitian ini bertujuan untuk mengembangkan sistem informasi penjualan berbasis

web yang mencakup fitur utama seperti manajemen pemesanan online, pemantauan stok berbasis, otomatisasi pencatatan transaksi melalui nota atau faktur digital, pemasaran berbasis web dengan fitur promosi digital, serta sistem manajemen kunjungan. Dengan implementasi sistem ini, efisiensi operasional diharapkan meningkat, jangkauan pasar menjadi lebih luas, serta pengalaman pelanggan dan pengunjung dapat lebih optimal.

Dari semua penelitian terdahulu yang telah dipaparkan, belum ada yang secara khusus mengintegrasikan pengelolaan stok, otomatisasi pencatatan transaksi, pemasaran digital, pengelolaan status pengiriman barang, serta manajemen kunjungan dalam satu sistem berbasis web seperti yang dirancang dalam penelitian ini. Keunggulan utama penelitian ini terletak pada pendekatannya yang mengombinasikan metodologi RSSD dan *Design Thinking*, yang memungkinkan sistem dikembangkan secara fleksibel, selaras dengan kebutuhan operasional, serta mampu meningkatkan efisiensi bisnis dan pengalaman pelanggan. Dengan menerapkan pendekatan *Design Thinking (Empathize, Define, Ideate)*, sistem dirancang berdasarkan kebutuhan pengguna, sementara RSSD memastikan proses pengembangan yang lebih cepat dan terstruktur dari perencanaan hingga pemeliharaan. Dengan integrasi metode ini, sistem informasi yang dihasilkan diharapkan dapat mengatasi keterlambatan transaksi, meningkatkan akurasi pencatatan, memperluas jangkauan pasar melalui pemasaran digital, mengoptimalkan pengelolaan pengiriman barang, serta mempermudah manajemen kunjungan, sehingga mendukung transformasi digital dan daya saing Central Palantea Hidroponik.

## **2.2 Landasan Teori**

### **2.2.1 Central Palantea Hidroponik**

Central Palantea hidroponik adalah sebuah perusahaan yang didirikan pada tahun 2019, yang berfokus pada produksi tanaman segar melalui metode hidroponik. Perusahaan ini memproduksi berbagai jenis tanaman seperti selada, pakcoy, tomat dan cabai. Selain berfungsi sebagai pusat produksi, Central Palantea juga menjadi wahana wisata dan edukasi yang menggabungkan ilmu pengetahuan, teknologi, dan lingkungan. Tempat ini menawarkan pengalaman belajar interaktif

tentang teknik budidaya tanaman tanpa tanah, yang sangat bermanfaat bagi berbagai kalangan, termasuk siswa dan pengunjung umum.



Gambar 2.1 Logo Central Palantea Hidroponik

### 2.2.2 *E-Commerce*

*E-commerce* merupakan seperangkat teknologi, aplikasi, dan proses bisnis yang bersifat dinamis, yang memungkinkan perusahaan, konsumen, dan komunitas tertentu untuk terhubung melalui transaksi secara elektronik. Aktivitas ini mencakup perdagangan barang, layanan, serta pertukaran informasi yang dilakukan melalui jaringan internet. Sementara itu, *e-business* mencakup kegiatan berbisnis di internet yang tidak terbatas pada aktivitas jual beli, tetapi juga mencakup pelayanan pelanggan serta kerja sama dengan mitra usaha, baik individu maupun lembaga.

Sebagai wujud perkembangan teknologi dalam bidang bisnis, *e-commerce* menjadi salah satu cara berbelanja secara *online* yang berkembang seiring meningkatnya penggunaan internet. Sistem ini memungkinkan transaksi pembelian dan penjualan dilakukan tanpa pertemuan langsung antara penjual dan pembeli, melainkan dengan memanfaatkan media komunikasi berbasis internet (Dasopang, 2023).



Gambar 2.2 E-commerce

### 2.2.3 Hidroponik

Hidroponik merupakan teknik budidaya pertanian yang dilakukan tanpa menggunakan media tanah, melainkan menggunakan air sebagai pengganti. Sistem tanam ini memungkinkan pemanfaatan lahan terbatas secara optimal. Dalam penerapannya, sistem hidroponik tidak membutuhkan lahan yang luas, sehingga cocok untuk diterapkan di area seperti pekarangan rumah, atap bangunan, maupun lahan sempit lainnya.

Keunggulan dari bercocok tanam secara hidroponik dibandingkan dengan metode konvensional yang menggunakan tanah antara lain terletak pada pengurangan gangguan hama dan penyakit, serta peningkatan kualitas hasil panen. Produk yang dihasilkan umumnya memiliki mutu yang lebih baik sehingga memiliki nilai jual yang lebih tinggi. Dengan demikian, keterbatasan lahan bukan lagi menjadi hambatan dalam upaya menciptakan lingkungan yang hijau dan menarik (Siregar & Novita, 2021).



Gambar 2.3 Hidroponik

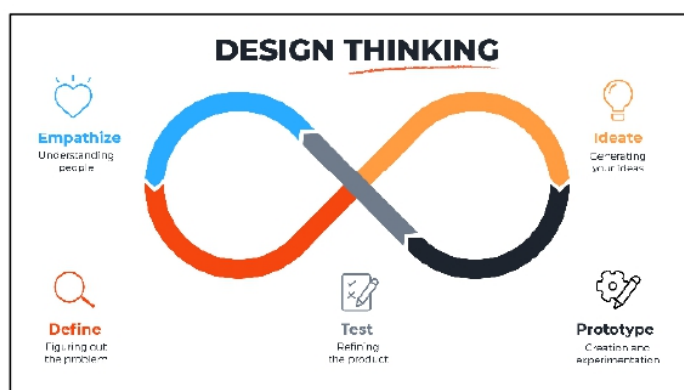
### 2.2.4 Design Thinking

Design Thinking merupakan pendekatan penyelesaian masalah yang berorientasi pada manusia (*human-centered approach*), di mana proses perancangan solusi dilakukan dengan menempatkan pengguna sebagai fokus utama. Pendekatan ini mengintegrasikan pemahaman terhadap kebutuhan pengguna, potensi teknologi, serta tujuan bisnis untuk menghasilkan solusi yang relevan dan aplikatif. Menurut Kelley dan Brown (2018), Design Thinking merupakan metode inovasi yang menekankan pada pemahaman mendalam terhadap kebutuhan manusia sebagai dasar dalam merancang solusi yang bernilai.

Dalam praktiknya, Design Thinking tidak hanya berfokus pada aspek kreativitas, tetapi juga mengombinasikan pendekatan sistematis dan analitis dalam merumuskan solusi yang efektif. Pendekatan ini bertujuan untuk menghasilkan solusi yang inovatif, efisien, serta berorientasi pada pengguna sebagai pusat dari proses perancangan (Fajrina & Postha, 2024).

Pada penelitian ini, Design Thinking diterapkan sebagai pendekatan awal untuk penggalan dan perumusan kebutuhan sistem sebelum memasuki tahapan pengembangan teknis. Penerapan Design Thinking difokuskan pada tiga tahapan utama, yaitu *empathize*, *define*, dan *ideate*. Tahap *empathize* digunakan untuk memahami kondisi operasional dan permasalahan yang dihadapi oleh pemilik usaha Central Palantea Hidroponik. Selanjutnya, tahap *define* digunakan untuk merumuskan permasalahan inti berdasarkan hasil pemahaman tersebut. Berdasarkan permasalahan yang telah dirumuskan, tahap *ideate* dimanfaatkan untuk menghasilkan gagasan solusi berupa sistem informasi penjualan berbasis web yang sesuai dengan kebutuhan pengguna dan tujuan usaha.

Penerapan ketiga tahapan Design Thinking ini bertujuan untuk memastikan bahwa sistem yang dikembangkan memiliki dasar kebutuhan yang jelas, relevan dengan kondisi pengguna, serta selaras dengan proses pengembangan sistem pada tahap selanjutnya.



Gambar 2.4 Tahapan Design Thinking

### 2.2.5 Metodologi Rapid Solo Software Development(RSSD)

RSSD merupakan metodologi pengembangan perangkat lunak yang dirancang khusus untuk pengembang tunggal, guna mengatasi keterbatasan kolaborasi dan sumber daya. Metodologi ini mengadopsi fase utama dari *Agile*, yaitu perencanaan, perancangan, pengembangan, pengujian, dan evaluasi. Namun,

RSSD menawarkan keunggulan tambahan berupa efisiensi proses, penerapan *modularitas*, serta kemudahan *revisitabilitas* atau peninjauan ulang kode secara berkelanjutan.

Efisiensi dalam RSSD dicapai melalui penyederhanaan tahap perencanaan dan struktur kode, sedangkan *modularitas* memastikan bahwa kode yang dikembangkan terdiri atas fungsi-fungsi kecil yang berdiri sendiri. Adapun *revisitabilitas* memungkinkan pengembang untuk memahami kembali struktur dan dokumentasi kode dengan lebih mudah pada masa mendatang. Hasil uji coba terhadap sepuluh proyek menunjukkan bahwa metodologi ini efektif dalam menyederhanakan proses pengembangan, dengan rata-rata tingkat kepuasan pengembang mencapai skor 4,19 dari skala 5,0 (Purba & Ramli, 2022).

Penerapan metodologi RSSD dalam pengembangan sistem dilakukan melalui beberapa tahapan utama. Berikut penjelasan untuk tiap tahapannya:

#### 1) Meet

Tahap *Meet* merupakan tahapan awal dalam metode RSSD yang berfungsi sebagai proses penyelarasan awal (*initial alignment*) antara pengembang dan pemangku kepentingan sebelum sistem direncanakan dan dikembangkan lebih lanjut. Tahap ini bertujuan untuk membangun pemahaman bersama mengenai konteks sistem, tujuan pengembangan, serta ruang lingkup permasalahan yang akan diselesaikan.

Berdasarkan konsep RSSD dalam literatur, tahap *Meet* tidak berfokus pada perancangan teknis, melainkan pada klarifikasi kebutuhan tingkat tinggi dan ekspektasi sistem. Aktivitas pada tahap ini digunakan untuk memastikan bahwa pengembangan sistem diarahkan pada permasalahan yang tepat dan sesuai dengan tujuan organisasi atau usaha yang bersangkutan.

Pada tahap *Meet*, pengembang dan stakeholder melakukan komunikasi awal untuk menyepakati gambaran umum sistem, peran pengguna, serta batasan pengembangan. Tahap ini menjadi fondasi penting dalam RSSD karena hasil dari *Meet* akan memengaruhi efektivitas perencanaan dan pengambilan keputusan pada tahap selanjutnya, khususnya *Planning & Pre-Evaluation*.

Dalam konteks RSSD yang bersifat cepat dan iteratif, tahap *Meet* juga berperan sebagai mekanisme untuk meminimalkan miskomunikasi sejak awal

pengembangan. Dengan adanya kesepahaman awal antara pengembang dan stakeholder, risiko perubahan kebutuhan secara signifikan pada tahap implementasi dapat dikurangi.

Tahap *Meet* dalam RSSD tidak menghasilkan artefak teknis secara langsung, namun menghasilkan kesepakatan konseptual yang menjadi dasar bagi proses perencanaan, evaluasi awal, serta pengembangan sistem pada tahapan berikutnya.

## 2) Planning & Pre-Evaluation

Tahap *Planning & Pre-Evaluation* dalam metode RSSD berfokus pada proses perencanaan teknis dan evaluasi awal sebelum sistem dirancang dan dikembangkan. Pada tahap ini, kebutuhan pengguna telah diperoleh dan divalidasi pada tahap sebelumnya, sehingga aktivitas pada RSSD difokuskan pada penyusunan rencana implementasi sistem secara terstruktur dan terukur.

Salah satu aktivitas utama pada tahap ini adalah re-evaluate, yaitu proses peninjauan ulang terhadap daftar kebutuhan dan fitur yang telah ditetapkan. *Re-evaluate* dilakukan untuk memastikan bahwa seluruh kebutuhan yang akan diimplementasikan relevan, realistis, dan dapat dikembangkan sesuai dengan batasan waktu serta sumber daya penelitian. Tahap ini berfungsi sebagai validasi awal sebelum sistem memasuki tahap perancangan teknis.

Setelah proses *re-evaluate*, dilakukan *chunking*, yaitu proses pengelompokan fitur ke dalam beberapa bagian (*chunk*) berdasarkan kesamaan fungsi dan keterkaitan proses bisnis. *Chunking* bertujuan untuk menyederhanakan kompleksitas sistem dengan membagi fitur menjadi modul-modul yang lebih kecil dan terstruktur. Dengan adanya pengelompokan ini, proses perancangan dan pengembangan sistem dapat dilakukan secara modular dan lebih mudah dikendalikan.

Hasil dari proses *chunking* kemudian dirangkum dalam bentuk final features list, yaitu daftar akhir fitur sistem yang akan dikembangkan. Daftar ini berfungsi sebagai acuan utama pada tahap *Design* dan *Develop*, sehingga pengembangan sistem tetap fokus pada fitur yang telah ditetapkan dan menghindari perubahan ruang lingkup yang tidak terkontrol.

Selain itu, tahap *Planning & Pre-Evaluation* juga mencakup penyusunan cost breakdown, yang merupakan estimasi kebutuhan sumber daya pengembangan

sistem. *Cost breakdown* digunakan untuk menggambarkan kebutuhan biaya secara konseptual, baik dari sisi pengembangan sistem maupun operasional awal, sehingga sistem yang dikembangkan tetap berada dalam batasan penelitian dan kebutuhan usaha.

Dalam kerangka RSSD yang bersifat iteratif, seluruh aktivitas pada tahap *Planning & Pre-Evaluation* berfungsi sebagai fondasi perencanaan teknis sebelum sistem dirancang dan dikembangkan. Tahap ini memastikan bahwa pengembangan sistem dilakukan secara terarah, terstruktur, dan siap untuk memasuki tahap *Design* tanpa risiko perubahan besar di tahap implementasi.

### 3) Design

Tahap *Design* merupakan tahapan perancangan sistem dalam metode RSSD yang bertujuan untuk menerjemahkan kebutuhan pengguna dan hasil analisis sebelumnya ke dalam bentuk rancangan sistem yang terstruktur. Tahap ini menjadi penghubung antara kebutuhan konseptual dan implementasi teknis sistem.

Pada tahap *Design*, fokus utama adalah menyusun gambaran sistem secara menyeluruh sebelum proses pengembangan kode dilakukan. Rancangan sistem dibuat untuk memastikan bahwa alur proses, struktur data, serta interaksi antar komponen sistem telah direncanakan dengan jelas dan dapat diimplementasikan secara efektif pada tahap *Develop*.

Aktivitas utama pada tahap *Design* meliputi penyusunan diagram perancangan sistem, seperti *Unified Modeling Language (UML)*, perancangan struktur data, serta perancangan alur proses bisnis sistem. Diagram UML digunakan untuk memvisualisasikan hubungan antar entitas, alur aktivitas pengguna, serta interaksi antara pengguna dan sistem. Dengan adanya perancangan ini, proses pengembangan dapat dilakukan secara lebih terarah dan terkontrol.

Selain itu, tahap *Design* juga mencakup perancangan struktur sistem dan pembagian modul berdasarkan fungsi dan peran pengguna. Perancangan ini bertujuan untuk memastikan bahwa sistem memiliki struktur yang jelas, mudah dipahami, serta mendukung pengembangan secara modular dan terintegrasi.

Dalam konteks RSSD yang bersifat iteratif, rancangan sistem pada tahap *Design* bersifat fleksibel dan dapat disesuaikan berdasarkan hasil pengujian dan evaluasi pada tahap selanjutnya. Dengan demikian, tahap *Design* tidak hanya berfungsi

sebagai pedoman pengembangan, tetapi juga sebagai acuan dalam melakukan penyempurnaan sistem secara berkelanjutan.

#### 4) Develop

Tahap *Develop* merupakan tahapan inti dalam metode RSSD yang berfokus pada proses implementasi sistem berdasarkan rancangan yang telah disusun pada tahap sebelumnya. Pada tahap ini, hasil perancangan diterjemahkan ke dalam bentuk kode program sehingga menghasilkan sistem yang dapat dijalankan dan digunakan.

Secara konseptual, tahap *Develop* bertujuan untuk mewujudkan solusi perangkat lunak yang fungsional melalui penerapan logika bisnis, pengelolaan data, serta penyusunan antarmuka sistem sesuai dengan kebutuhan pengguna. Proses pengembangan dilakukan secara terstruktur dan modular agar setiap komponen sistem dapat dikembangkan, diuji, dan diintegrasikan dengan baik.

Aktivitas utama pada tahap *Develop* meliputi implementasi kode program (*code body*), integrasi antar modul (*integrate code*), serta verifikasi awal sistem (*verify the code*). Implementasi kode program dilakukan dengan membangun fitur-fitur utama sistem sesuai dengan spesifikasi yang telah dirancang. Setiap fitur dikembangkan berdasarkan peran pengguna dan alur proses yang telah ditentukan, sehingga sistem mampu mendukung kebutuhan operasional secara menyeluruh.

Tahap integrasi kode dilakukan untuk memastikan bahwa seluruh modul yang telah dikembangkan dapat saling berinteraksi dengan baik. Integrasi ini mencakup keterhubungan antara modul pengguna, pengelolaan data, serta proses bisnis utama, sehingga alur sistem dapat berjalan secara utuh tanpa hambatan fungsional.

Selanjutnya, dilakukan verifikasi awal terhadap sistem untuk memastikan bahwa hasil implementasi berada dalam kondisi siap diuji. Verifikasi difokuskan pada pemeriksaan alur proses utama, akses fitur berdasarkan peran pengguna, serta kestabilan sistem secara umum. Tahap ini tidak bertujuan untuk menemukan kesalahan secara rinci, melainkan untuk memastikan bahwa sistem telah siap memasuki tahap pengujian formal.

Dalam kerangka RSSD yang bersifat iteratif, tahap *Develop* tidak berdiri sendiri, melainkan dapat dilakukan secara berulang berdasarkan hasil pengujian dan evaluasi. Dengan demikian, tahap *Develop* berperan penting dalam menghasilkan

sistem yang fungsional, terintegrasi, dan siap untuk diuji serta dievaluasi pada tahap selanjutnya.

#### 5) Test

Tahap *Test* merupakan tahapan pengujian sistem dalam metode RSSD yang bertujuan untuk memastikan bahwa sistem yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan untuk memverifikasi bahwa setiap fungsi utama sistem menghasilkan keluaran yang sesuai dengan masukan yang diberikan, sehingga sistem dapat digunakan secara andal dalam mendukung proses bisnis.

Pada tahap *Test*, fokus pengujian diarahkan pada fungsionalitas sistem dan kestabilan alur proses, bukan pada struktur internal kode program. Oleh karena itu, metode pengujian yang umum digunakan pada tahap ini adalah *Black Box Testing*. Metode ini memungkinkan pengujian dilakukan dari sudut pandang pengguna dengan mengevaluasi kesesuaian antara input dan output tanpa mempertimbangkan implementasi teknis di dalam sistem.

Aktivitas utama pada tahap *Test* meliputi penyusunan skenario pengujian, pelaksanaan pengujian terhadap fitur-fitur utama sistem, serta pencatatan hasil pengujian dalam bentuk *test report*. Setiap skenario pengujian mencakup masukan (input), keluaran yang diharapkan (*expected output*), keluaran aktual (*actual output*), serta status hasil pengujian. Hasil pengujian digunakan untuk memastikan bahwa seluruh fungsi sistem telah berjalan sesuai dengan kebutuhan pengguna.

Selain pengujian fungsional, tahap *Test* juga mencakup proses optimasi sistem (*optimize code*). Optimasi dilakukan untuk meningkatkan efisiensi, kestabilan, dan keamanan sistem tanpa mengubah fungsi utama yang telah dirancang pada tahap sebelumnya. Proses ini bertujuan untuk meminimalkan potensi kesalahan, meningkatkan kenyamanan penggunaan, serta memastikan sistem berada dalam kondisi siap untuk dievaluasi oleh pengguna.

Dalam konteks RSSD yang bersifat iteratif, hasil dari tahap *Test* menjadi dasar untuk melanjutkan sistem ke tahap *Evaluate*. Sistem yang telah melewati tahap *Test* diharapkan berada dalam kondisi stabil secara fungsional, sehingga proses evaluasi selanjutnya dapat difokuskan pada penerimaan pengguna dan kesesuaian sistem dengan kebutuhan operasional.

## 6) Evaluate

Tahap *Evaluate* merupakan tahapan evaluasi dalam metode RSSD yang bertujuan untuk menilai kesesuaian sistem yang telah dikembangkan dengan kebutuhan pengguna dan tujuan bisnis. Tahap ini dilakukan setelah proses *develop* dan *test* untuk memastikan bahwa sistem tidak hanya berfungsi secara teknis, tetapi juga dapat digunakan secara efektif dan diterima oleh pengguna.

Pada tahap *Evaluate*, fokus utama evaluasi tidak lagi berada pada struktur kode atau implementasi teknis, melainkan pada validasi sistem dari sudut pandang pengguna. Evaluasi dilakukan untuk menilai apakah alur penggunaan sistem telah sesuai, fitur-fitur utama dapat mendukung proses bisnis, serta sistem memberikan pengalaman penggunaan (*usability*) yang baik.

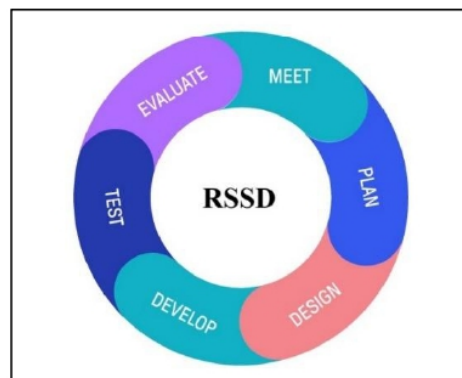
Secara konseptual, tahap *Evaluate* dalam RSSD mencakup beberapa aktivitas utama, yaitu diskusi dengan pengguna, demonstrasi sistem, dan pengumpulan umpan balik (*feedback*). Diskusi dilakukan untuk memperoleh pemahaman langsung mengenai kesesuaian sistem dengan kebutuhan operasional pengguna. Demonstrasi sistem bertujuan untuk memperlihatkan cara kerja sistem secara menyeluruh dan memastikan bahwa alur proses utama dapat dipahami dan dijalankan dengan baik oleh pengguna. Sementara itu, pengumpulan *feedback* dilakukan sebagai bentuk validasi akhir terhadap sistem yang dikembangkan.

Selain *feedback* kualitatif, tahap *Evaluate* juga dapat dilengkapi dengan evaluasi kuantitatif untuk mengukur tingkat kemudahan penggunaan sistem. Salah satu metode yang umum digunakan adalah *System Usability Scale (SUS)*, yang berfungsi untuk menilai *usability* sistem secara terstandar berdasarkan persepsi pengguna. Hasil evaluasi ini digunakan untuk menentukan tingkat penerimaan sistem serta mengidentifikasi potensi penyempurnaan sebelum sistem memasuki tahap pemeliharaan (*maintenance*).

Dalam konteks RSSD yang bersifat iteratif, hasil dari tahap *Evaluate* tidak hanya berfungsi sebagai penilaian akhir, tetapi juga menjadi dasar untuk melakukan perbaikan (*fix bugs*) dan penyempurnaan fitur (*change features as needed*) pada tahap *Maintenance*. Dengan demikian, tahap *Evaluate* berperan

penting dalam memastikan bahwa sistem yang dikembangkan telah layak digunakan secara operasional dan siap untuk diimplementasikan.

Pengembangan sistem dilanjutkan pada tahap *develop* yang berisi proses penulisan dan integrasi kode secara modular, diikuti dengan pengujian fungsionalitas sistem dalam tahap *test* untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan. Evaluasi dilakukan melalui demonstrasi sistem kepada pemilik usaha dan pengumpulan umpan balik menggunakan metode *System Usability Scale* (SUS). Terakhir, tahap *maintenance* dijalankan secara berkelanjutan, termasuk perbaikan bug dan pembaruan fitur agar sistem tetap optimal dan relevan terhadap kebutuhan pengguna.



Gambar 2.5 Tahapan Rapid Solo Software Development(RSSD)

### 2.2.6 *Laravel*

*Laravel* merupakan *framework* pengembangan aplikasi web berbasis *PHP* versi 5.3 yang bersifat *open-source*, dan dikembangkan oleh Taylor Otwell dengan lisensi di bawah *MIT License*. Tujuan utama dari *Laravel* adalah mempermudah proses pengembangan aplikasi dengan menyajikan sintaks yang sederhana, elegan, dan ekspresif, sehingga menciptakan pengalaman pengkodean yang menyenangkan dan efisien.

Dengan menggunakan *Laravel*, pengembang dapat mengurangi kompleksitas dalam proyek melalui fitur-fitur seperti *routing*, *session*, dan *caching* yang sudah tersedia. Selain itu, *Laravel* juga mengadopsi prinsip dan pengalaman dari berbagai bahasa pemrograman dan *framework* lainnya, seperti *Ruby on Rails*, *ASP.NET*, *MVC*, dan *Sinatra*. Hal ini menjadikan *Laravel* sebagai salah satu *framework* yang fleksibel dan mudah diintegrasikan dalam berbagai kebutuhan pengembangan aplikasi web (Nugraha, 2014).

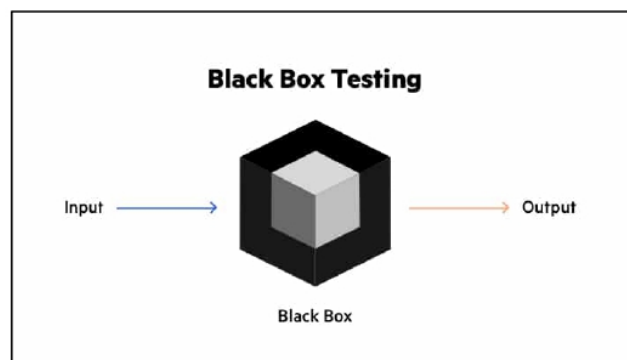


Gambar 2.6 Laravel

### 2.2.7 *Blackbox Testing*

Metode *Blackbox testing* merupakan metode pengujian perangkat lunak yang difokuskan pada aspek fungsional dari sistem tanpa melihat struktur internal atau kode program. Pengujian ini bertujuan untuk menemukan berbagai kesalahan, seperti fungsi yang tidak berjalan sebagaimana mestinya, kesalahan pada antarmuka pengguna (*interface error*), kesalahan dalam struktur maupun akses basis data, performa sistem (*performance error*), serta kesalahan dalam proses inisialisasi dan terminasi.

Teknik pengujian ini mencakup pendekatan seperti *equivalence partitioning*, *boundary value analysis*, dan *limit testing*. Pendekatan tersebut digunakan untuk memastikan bahwa setiap fitur telah berfungsi sesuai dengan kebutuhan yang telah dirancang sebelumnya. Metode *blackbox testing* sering digunakan karena mampu menghasilkan dokumentasi pengujian yang menggambarkan kesesuaian perangkat lunak terhadap spesifikasi sistem yang telah ditentukan (Achmad & Yulfitri, 2020).

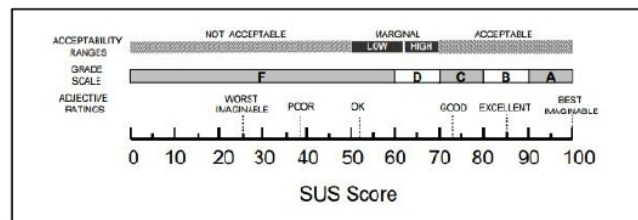


Gambar 2.7 Blackbox Testing

### 2.2.8 System Usability Scale (SUS)

SUS merupakan metode evaluasi yang banyak digunakan untuk menilai tingkat *usability* atau kemudahan penggunaan suatu antarmuka sistem, baik berupa aplikasi maupun situs web. Metode ini mengandalkan kuesioner standar yang terdiri atas sepuluh pernyataan, masing-masing diberi penilaian dari skala 1 hingga 5. Pernyataan bernomor ganjil digunakan untuk mengukur persepsi positif pengguna, sedangkan pernyataan bernomor genap mengevaluasi potensi kesulitan atau kendala yang dihadapi.

Perhitungan skor dilakukan dengan mengurangi 1 dari skor pernyataan ganjil dan mengurangi skor pernyataan genap dari angka 5. Hasil dari setiap peserta kemudian dijumlahkan dan dikalikan dengan 2,5 sehingga menghasilkan skor akhir dalam rentang 0 hingga 100. Skor ini menggambarkan kualitas *usability* sistem, di mana nilai lebih tinggi menunjukkan kemudahan penggunaan yang lebih baik. Berdasarkan interpretasi standar *SUS*, nilai di atas 70 mengindikasikan tingkat kepuasan pengguna yang baik dan mendekati kategori “excellent” (Chandra & Zul, 2024).



Gambar 2.8 System Usability Scale (SUS)

### 2.2.9 Database

*Database* merupakan kumpulan data yang disimpan secara sistematis dalam sistem komputer untuk mendukung proses pengolahan dan manajemen informasi. Data yang tersimpan dapat berupa teks, gambar, video, atau jenis file lainnya. Sistem *database* menggunakan perangkat lunak untuk menyimpan, mengelola, memindahkan, dan mengedit data dengan cepat dan efisien. Dengan struktur yang terorganisasi, *database* memungkinkan penyimpanan informasi yang bersifat permanen serta mempermudah pengambilan data untuk berbagai keperluan organisasi.

*Database* berfungsi sebagai pusat penyimpanan data digital yang dapat dimanfaatkan untuk keperluan administrasi, pengolahan data operasional, hingga

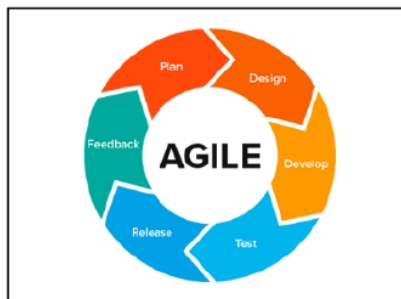
pendukung keputusan. Informasi yang disimpan disusun secara sistematis agar dapat diakses kembali dengan mudah dan tidak terjadi manipulasi terhadap sistem. Umumnya, *database* juga digunakan untuk mempermudah pemakai dalam mengelola sistem secara efisien dan terstruktur, baik dalam skala perusahaan maupun organisasi lainnya. Selain itu, tinjauan pustaka (*literature review*) dan dokumentasi manual sistem turut mendukung pemahaman pengguna terhadap abstraksi data yang telah disediakan (Pulungan dkk., 2022).



Gambar 2.9 Database

#### 2.2.10 Agile

*Agile* merupakan pendekatan manajemen yang menekankan fleksibilitas, kecepatan adaptasi, serta kemampuan organisasi dalam merespons perubahan secara efektif di tengah lingkungan yang dinamis dan penuh ketidakpastian. Pendekatan ini mendorong kolaborasi yang kuat antar tim, pengambilan keputusan yang cepat berbasis data, serta proses kerja yang bersifat iteratif dan berkelanjutan. Dalam konteks organisasi modern, Agile tidak hanya berfokus pada efisiensi proses, tetapi juga pada pengembangan sumber daya manusia yang adaptif, inovatif, dan responsif terhadap perubahan kebutuhan bisnis. Dengan demikian, penerapan Agile dapat membantu organisasi dalam meningkatkan kinerja, mempercepat pencapaian tujuan, serta menciptakan keunggulan kompetitif di era digital (Silalahi dkk., 2025).



Gambar 2.10 Agile