

SIMAGA (Sistem Informasi Master Gambar) Untuk Pengelolaan Stok Gambar Pada CV. Berkah Nusantara Jaya

Ahmad Zulfikar¹, Rin Megawatty², Nabil Fahri Rahmatullah³, Rudianto⁴

^{1,2}Universitas Bina Sarana Informatika
Jl. Kramat Raya, Kec. Senen, Jakarta, Indonesia

email korespondensi: tanahbeta1945@gmail.com

Abstrak

Era digital membawa perkembangan pesat dalam teknologi informasi dan telekomunikasi, mengubah cara hidup dan bekerja. Pertukaran data pun semakin intensif, termasuk data gambar desain yang menjadi aset berharga bagi perusahaan. Sebuah perusahaan CV Berkah Nusantara Jaya, masih memiliki sistem pengelolaan data gambar yang tidak terpusat, sehingga menyulitkan pencarian, akses, dan berbagi gambar. Hal ini mendorong kebutuhan akan sistem informasi terpusat untuk mengelola dan menyimpan data gambar desain secara efisien dan mudah diakses. Penelitian ini bertujuan untuk mengembangkan sistem informasi pengelolaan data gambar desain berbasis web untuk CV Berkah Nusantara Jaya. Untuk pengembangan aplikasi digunakan metode *waterfall*. Metode *waterfall* ini adalah salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model *waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *waterfall* antara lain *analysis, design, implementation, testing, maintenance*. Dengan pengembangan aplikasi sistem informasi ini diharapkan dapat meningkatkan efisiensi dan aksesibilitas data gambar, sehingga mendukung kelancaran operasional perusahaan dan meningkatkan daya saingnya.

Kata kunci: pengelolaan data gambar, sistem informasi, CV. Berkah Nusantara Jaya, efisiensi, aksesibilitas.

Abstracts

The digital era has brought rapid developments in information technology and telecommunications, changing the way we live and work. Data exchange is increasingly intensive, including design drawing data which has become a valuable asset for companies. A company, CV Berkah Nusantara Jaya, still has a non-centralized image data management system, making it difficult to search, access and share images. This drives the need for a centralized information system to manage and store design drawing data efficiently and easily accessed. This research aims to develop a web-based design image data management information system for CV Berkah Nusantara Jaya. For application development, the waterfall method is used. This waterfall method is an SDLC model that is often used in developing information systems or software. The waterfall model uses a systematic and sequential approach. The stages of the waterfall model include analysis, design, implementation, testing, maintenance. By developing this information system application, it is hoped that it can increase the efficiency and accessibility of image data, thereby supporting the company's smooth operations and increasing its competitiveness.

Keywords: design data management, information system, CV. Berkah Nusantara Jaya, efficiency, accessibility.

1. Pendahuluan

Perkembangan teknologi saat ini sangat berkembang pesat dalam perkembangan manusia sepanjang sejarah. Di era tahun 2000 adalah puncak kemajuan teknologi yang sangat pesat perkembangannya, dimana teknologi informasi dan telekomunikasi menjadi *trend* kehidupan setiap individu, tiap saat, tiap waktu dan tiap detik manusia memanfaatkan teknologi ini (Danuri, 2019). Teknologi saat ini terus berkembang, mengubah cara kita hidup, bekerja hingga berkomunikasi.

Pertukaran data dan informasi pun semakin bertambah. Data yang dahulu hanya bisa diolah perangkat lunak yang terbatas kini bukan lagi jadi masalah. Pemrosesan data bisa dilakukan dengan menggunakan perangkat lunak apa saja sesuai dengan kebutuhan. Seiring dengan berkembangnya teknologi dalam pengarsipan data itu, semakin besar pula ukuran dari data yang diarsipkan. Keberadaan sebuah arsip bagi suatu organisasi merupakan hal yang penting, karena di dalam arsip tersebut terkandung informasi-informasi yang penting dan memiliki nilai guna bagi organisasi (Bahrudin et al., 2020). Oleh karena itu, proses pengarsipan dokumen dalam sebuah perusahaan menjadi hal yang penting karena di dalam arsip dokumen itu terdapat data-data penting yang menjadi



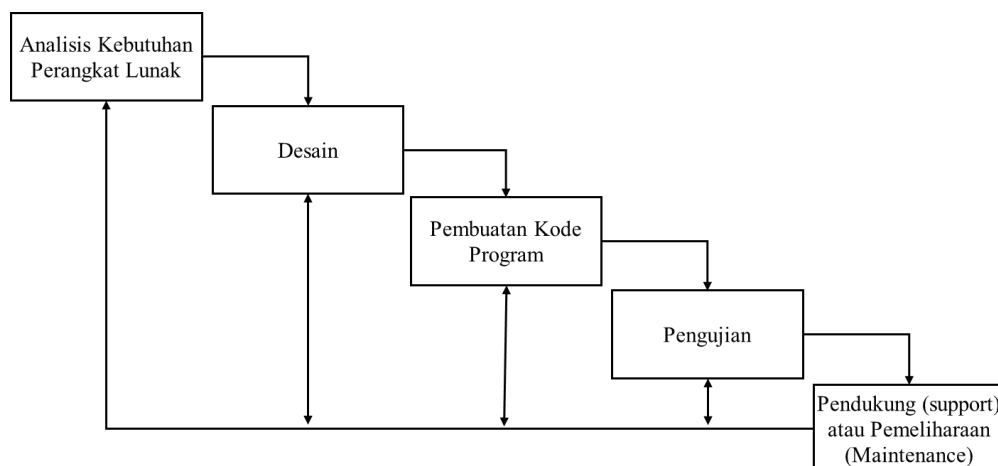
pondasi perusahaan untuk dapat bertahan dan juga berkembang. Salah satu contoh dokumen penting dalam sebuah perusahaan adalah desain gambar sebuah produk.

Gambar merupakan sebuah alat untuk menyatakan maksud, terutama bagi orang-orang teknik. Oleh karena itu gambar sering juga disebut sebagai bahasa Teknik. Sebagai bahasa teknik, diharapkan sebuah gambar dapat meneruskan keterangan-keterangan secara tepat dan obyektif (Fathun, 2020). Desain gambar yang dibuat oleh para pembuatnya atau disebut juga sebagai *drafter*, terus berkembang menyesuaikan dengan kebutuhan dari *customer*, sehingga menghasilkan dokumen-dokumen desain gambar yang banyak dan dapat dijadikan pula sebagai desain gambar dasar untuk dimodifikasi atau dikembangkan menjadi produk yang lebih sesuai dengan kebutuhan atau keinginan *customer*.

Banyaknya dokumen desain gambar kendaraan yang telah dibuat oleh para *drafter* menjadi stok data gambar yang berharga bagi perusahaan. Hal ini harus menjadikan pengelolaan dan distribusi data gambar menjadi bagian penting untuk meningkatkan kinerja perusahaan. Sehingga perusahaan tersebut memerlukan penggunaan sebuah sistem informasi untuk mengelola dan menyimpan gambar yang dapat meningkatkan efisiensi dan aksesibilitas dalam mengelola stok gambar yang ada. Pengelolaan data barang pada suatu perusahaan adalah masalah yang krusial dan harus ditangani dengan seksama. Karena hal ini akan mempengaruhi siklus kelangsungan hidup pada perusahaan tersebut. Dalam bidang informatika permasalahan ini dapat diselesaikan dengan teori dalam bidang informatika, khususnya dengan metode *searching* (Ardi Widodo et al., 2021). Saat ini pengelolaan data gambar desain pada CV. Berkah Nusantara Jaya masih belum tersimpan secara terpusat atau masih menyimpan pada masing-masing individu *drafter*. Hal tersebut menyebabkan kesulitan dalam mencari, mengakses dan berbagi gambar. Selain itu, ketika jumlah gambar meningkat, pencarian gambar menjadi semakin sulit. Tanpa sistem yang terstruktur, pengguna sering kali menghabiskan waktu untuk mencari gambar yang diperlukan.

2. Metode

Dalam penelitian ini, penulis menggunakan pengembangan *System Development Life Cycle* (SDLC). SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem (Rahmi et al., 2023). SDLC terdiri dari beberapa fase yang dimulai dari fase perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem. Metode *waterfall* adalah salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Pada prosesnya tahapan yang dilakukan adalah satu per satu diselesaikan terlebih dahulu kemudian melangkah pada tahap berikutnya setelah sepenuhnya selesai (Yulia et al., 2021). Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (maintenance) dan dilakukan secara bertahap. (Wahid, 2020). Berikut tahapan dari metode *waterfall* dapat dilihat pada ada gambar di bawah ini:



Sumber : Ardi Widodo (2021)

Gambar 1. Metode *Waterfall*

1. Analisis (*Analysis*) Kebutuhan Perangkat Lunak

Pada tahap ini pengembang harus mengetahui seluruh informasi mengenai kebutuhan *software* seperti kegunaan *software* yang diinginkan oleh pengguna dan batasan *software*. Informasi tersebut biasanya diperoleh dari wawancara, survey, ataupun diskusi. Setelah itu *informs* dianalisis sehingga mendapatkan data-data yang lengkap mengenai kebutuhan pengguna akan *software* yang akan dikembangkan.

2. Desain (*Design*)

Tahap selanjutnya yaitu Desain. Desain dilakukan sebelum proses *coding* dimulai. Ini bertujuan untuk memberikan gambaran lengkap tentang apa yang harus dikerjakan dan bagaimana tampilan dari sebuah sistem yang diinginkan. Sehingga membantu menspesifikasikan kebutuhan *hardware* dan sistem, juga mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan.

3. Pembuatan Kode Program (*Implementation*)

Proses penulisan *code* ada di tahap ini. Pembuatan *software* akan dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

4. Pengujian (*Testing*)

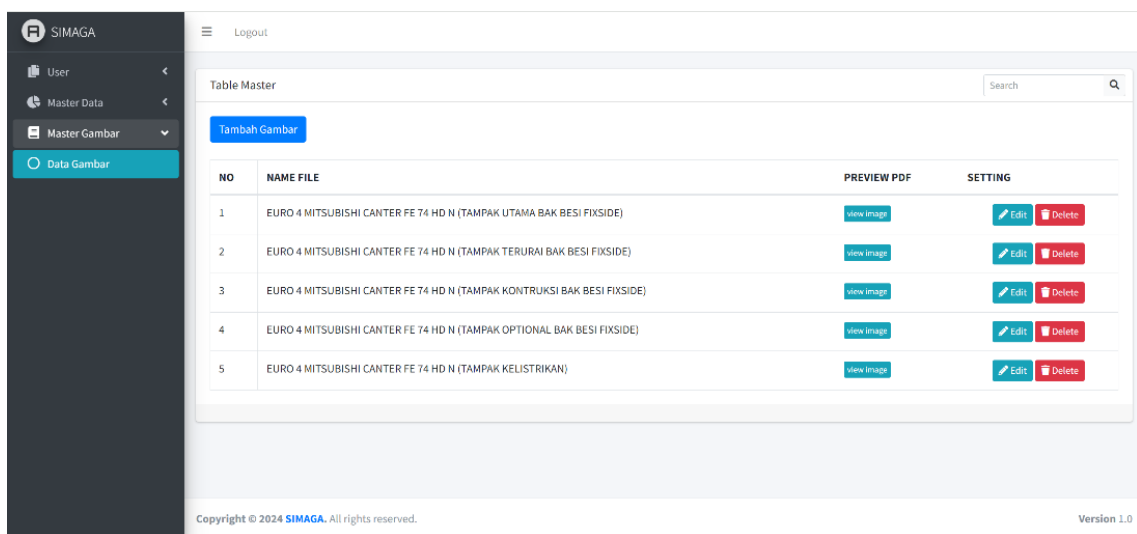
Pada tahap keempat ini akan dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya. Setelah itu akan dilakukan pengujian yang bertujuan untuk mengetahui apakah *software* sudah sesuai desain yang diinginkan dan apakah masih ada kesalahan atau tidak.

5. Pemeliharaan (*Maintenance*)

Maintenance adalah tahapan terakhir dari metode pengembangan *waterfall*. Di sini *software* yang sudah jadi akan dijalankan atau dioperasikan oleh penggunanya. Di samping itu dilakukan pula pemeliharaan yang termasuk : perbaikan kesalahan, perbaikan implementasi unit sistem, peningkatan jasa sistem sesuai kebutuhan baru (Supiyandi et al., 2022)

3. Hasil dan Pembahasan

Berdasarkan penelitian yang telah dilakukan terhadap CV. Berkah Jaya Nusantara diketahui bahwa perusahaan tersebut membutuhkan sebuah sistem untuk membantu pengelolaan stok gambar. Hasilnya kami membuat sebuah aplikasi bernama SIMAGA (Sistem Informasi Master Gambar) yaitu sebuah aplikasi sistem informasi pengelolaan stok gambar yang dapat membantu antar pengguna dalam menyimpan dan mengelola stok gambar secara terpusat, terintegrasi, efisien dan dapat diakses oleh banyak pengguna.



Gambar 2. Tampilan Aplikasi SIMAGA

A. Analisa kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak adalah proses yang penting dalam pengembangan perangkat lunak untuk memahami dan mendokumentasikan kebutuhan *user*. Proses ini memastikan bahwa perangkat lunak yang dikembangkan memenuhi harapan dan kebutuhan yang ditetapkan

1. Analisa Kebutuhan Sistem

Untuk membangun sebuah aplikasi perlu diketahui kebutuhan dan keinginan dari *user*. Yaitu apa yang diharapkan dari *user* dari aplikasi yang dibuat. Pada tahapan ini, diketahui *user* menginginkan aplikasi yang dapat mengelola data gambar desain mobil. Hingga akhirnya diputuskan untuk membuat sebuah aplikasi dengan nama SIMAGA (Sistem Informasi Master Gambar).

2. Analisa Kebutuhan Fungsional

Analisa kebutuhan fungsional merupakan gambaran dari proses-proses mengenai sistem yang berjalan pada sistem SIMAGA. Adapun beberapa fungsi yang harus dapat dijalankan pada SIMAGA adalah :

- Sistem ini memiliki 2 macam pengguna, yaitu sebagai admin dan sebagai *drafter*
- Sebagai Admin, pengguna aplikasi dapat melakukan :
 - a. Mengelola data login *user* dan *role* dari masing-masing *user*
 - b. Mengelola data Pegawai
 - c. Mengelola data *Customer*
 - d. Mengelola data Gambar Desain
 - e. Meng-*upload* Gambar Desain
- Sebagai *Drafter*, pengguna aplikasi dapat melakukan :
 - a. Mencari dan melihat data stok gambar desain
 - b. Melihat dan mengambil data gambar desain

3. Kebutuhan Non Fungsional

Analisa kebutuhan non-fungsional merupakan analisa yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Pada analisis kebutuhan non fungsional dijelaskan analisis kebutuhan perangkat keras, perangkat lunak, dan juga analisis pengguna diantaranya sebagai berikut:

1. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang diperlukan dalam pembuatan website “SIMAGA” adalah:

- a. *Web browser* (Google Chrome, Mozilla, Edge)
- b. Internet

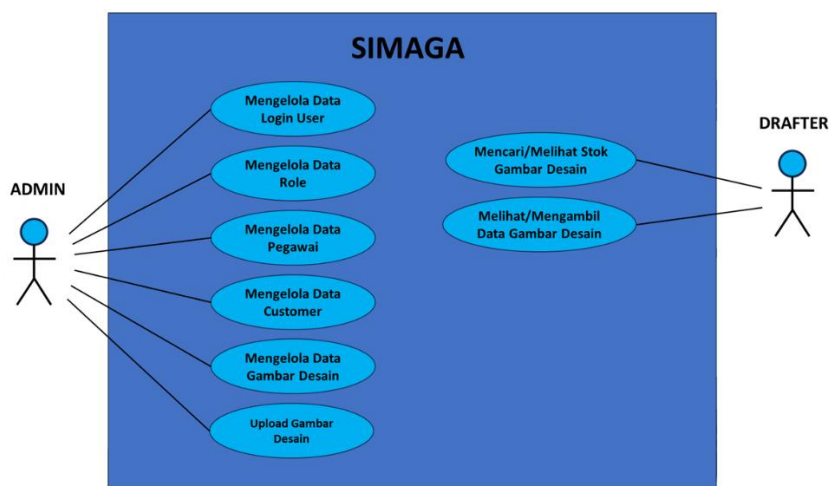
2. Perangkat Keras (*Hardware*)

- a. Komputer
- b. Monitor
- c. *Mouse*
- d. *Keyboard*
- e. *Wifi*

B. Desain

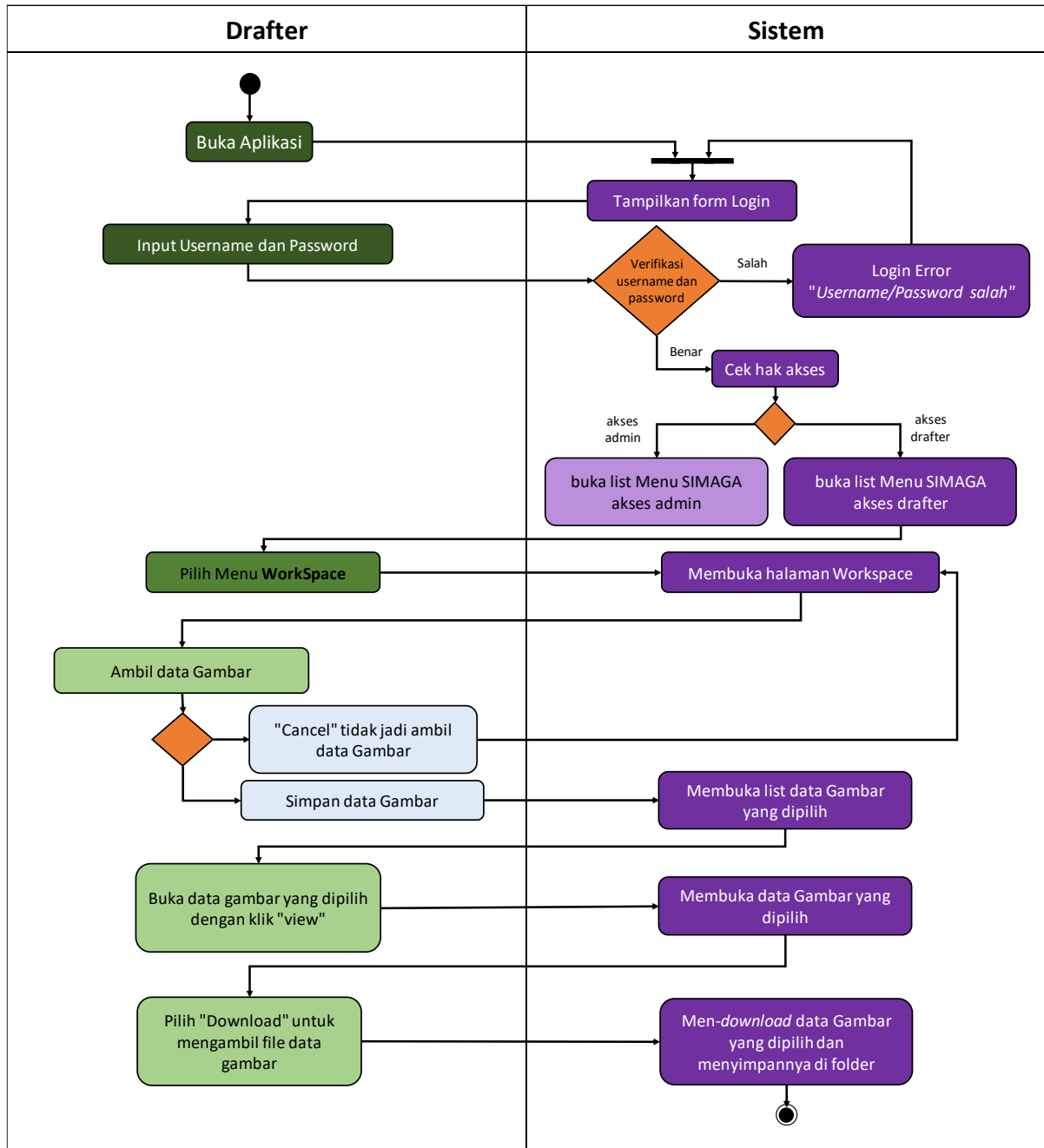
Unified Modeling Language (UML) adalah bahasa untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu, UML juga merupakan Bahasa pemodelan yang menggunakan konsep orientasi object (Aurin et al., 2021). Untuk memvisualisasikan sistem aplikasi yang dijalankan penulis menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Berikut hasil pembuatan diagramnya :

1. *Use Case Diagram*



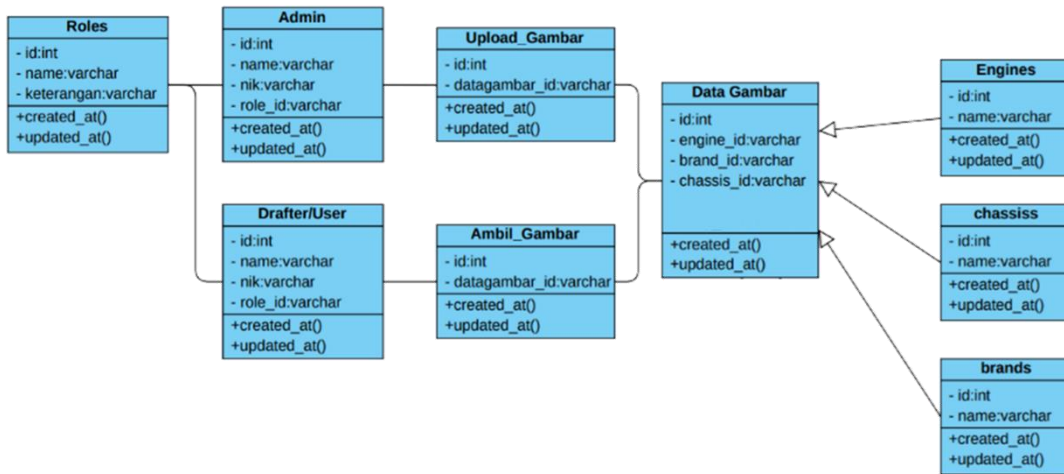
Gambar 3. *Use Case* aplikasi SIMAGA

2. Activity Diagram



Gambar 4. Salah satu Activity Diagram admin dalam aplikasi SIMAGA

3. Class Diagram



Gambar 5. Class Diagram aplikasi SIMAGA

C. Pembuatan Kode Program

Untuk pembuatan kode program aplikasi yang di bangun, penulis menggunakan *framework Laravel*. Pada pengkodean ini menggunakan *MVC (Model,View,Controller)* dan *Route*. Berikut beberapa kode aplikasi yang telah dibangun:

1. Model

```

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'role_id',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     */
}
    
```

Gambar 6. List Program Model

2. View


```

3 s="content-wrapper">
5 ion class="content">
6 div class="container-fluid">
7 <div class="row">
8 <div class="col-12">
44
45 @if (empty($masters))
46 @foreach ($masters as $master)
47 <td>{{ $loop->iteration }}</td>
48 <td>{{ $master->connections->id ?? '' }}</td>
49 <td>
50 <button type="button" class="btn btn-xs btn-info" onclick="$ .get
51 | data-toggle="lightbox"> view image</button>
52 </td>
53 <td class="project-actions text-center">
54 <a class="btn btn-info btn-sm"
55 | href="{{ route('edit.master', ['id' => $master->id]) }}"><i
56 | class="fas fa-pencil-alt"></i> Edit</a>
57 <a class="btn btn-danger btn-sm"
58 | href="{{ route('delete.master', ['id' => $master->id]) }}"><i
59 | class="fas fa-trash"></i> Delete</a>
60 </td>

```

Gambar 7. List Program View

3. Controller

```

15
16 class UserController extends Controller
17 {
18 public function index() {
19     $data = ['title'=>'Data User Login',
20 | | | | 'users'=>$this->view(),
21 | | | | ];
22     return view('backend.users.index_user',$data);
23 }
24
25 public function view () {
26     try {
27         $data = User::paginate();
28         return $data;
29     }
30     catch(Exception $e){
31         dd($e->getMessage());
32     }
33 }

```

Gambar 8. List Program Controller

4. Route

```

36
37
38 Route::controller(LoginAdminController::class)->group(function () {
39     //Controller View login
40     | Route::get('/', 'index')->name('login.panel');
41     //Controller View registrasi
42     | Route::get('/register', 'register')->name('login.register');
43     // route for function controller save registrasi
44     | Route::post('/register-save', 'register_save')->name('login.register.save');
45     // route for function controller logout
46     | Route::get('/logout', 'logout')->name('logout');
47     // route for function controller auth
48     | Route::get('/login', 'auth')->name('login');
49     });
50
51 Route::controller(HomeController::class)->group(function () {
52     //Controller View home Admin
53     | Route::get('/home', 'index')->name('home');
54     });
55
56 Route::middleware(['auth', 'auth.session'])->group(function () {

```

Gambar 9. List Program Route

D. Pengujian

Pengujian dilakukan menggunakan *Black Box*. *Black Box* testing yang berarti bahwa pengujian suatu aplikasi harus terdapat pada inputan dan output atau keluaran atas penilaian suatu aplikasi dan tidak terdapat pada fungsi dari aplikasi atau spesifikasi logikanya saja (Abdilah et al., n.d.). berikut ini terdapat beberapa pengujian pada aplikasi yang dibangun :

1. Halaman Admin

Table 1. Skenario Pengujian Halaman Admin

No	Komponen Pengujian	Skenario Pengujian	Jenis Pengujian
1.	Login Admin	Aplikasi menampilkan tampilan login yang nantinya akan digunakan oleh admin	<i>Black Box</i>
2.	Logout Admin	Menampilkan form <i>Login</i>	<i>Black Box</i>
3.	<i>User</i>	Menampilkan Submenu yang berisi data login, Data <i>Role</i> , Data Pegawai, Data <i>Customer</i>	<i>Black Box</i>
4.	Submenu Data <i>Login</i> , Data <i>Role</i> , Data Pegawai, Data <i>Customer</i>	Menampilkan Data Submenu. Selain itu pengguna juga dapat menambah, edit dan <i>delete</i> data.	<i>Black Box</i>
5.	Master Data	Menampilkan Submenu yang berisi <i>type engine</i> , <i>type merk</i> , <i>type chassis</i> , <i>type vehicle</i> , varian body, <i>type varian</i> , <i>type submission</i> , relasi master data.	<i>Black Box</i>
6.	Sub Menu <i>Type Engine</i> , <i>type merk</i> , <i>type chassis</i> , <i>type vehicle</i> , varian body, <i>type varian</i> , <i>type submission</i> , relasi master data.	Menampilkan data submenu. Selain itu pengguna juga dapat menambah, edit dan <i>delete</i> data.	<i>Black Box</i>
7.	Master Gambar	Menampilkan Submenu yang berisi Tampak Utama, Tampak Terurai, Tampak Kontruksi, Tampak Optional, Tampak Kelistrikan.	<i>Black Box</i>
8.	Sub Menu Tampak Utama, Tampak Terurai, Tampak Kontruksi, Tampak Optional, Tampak Kelistrikan.	Menampilkan data submenu. Selain itu pengguna juga dapat menambah, edit dan <i>delete</i> data.	<i>Black Box</i>

2. Halaman *Drafter*

Table 2. Skenario Pengujian Halaman *Drafter*

No	Komponen Pengujian	Skenario Pengujian	Jenis Pengujian
1.	<i>Login</i>	Aplikasi menampilkan tampilan <i>login</i> yang nantinya akan digunakan oleh <i>drafter</i>	<i>Black Box</i>
2.	<i>Logout</i>	Menampilkan form <i>Login</i>	<i>Black Box</i>
3.	<i>WorkSpace</i>	Menampilkan data lembar kerja. Selain itu pengguna juga dapat menambah, edit dan <i>delete</i> data	<i>Black Box</i>

Berikut ini adalah hasil dari pengujian dari skenario penggunaan aplikasi SIMAGA dengan menggunakan *Black Box* :

Table 3. Hasil Pengujian Halaman Admin

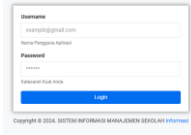
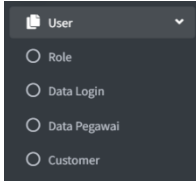
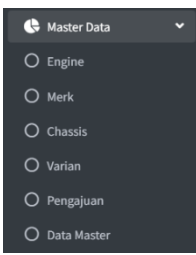
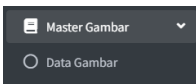

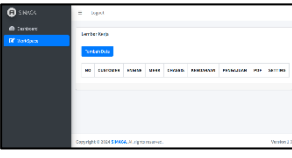
NO	Komponen yang diuji	Skenario dan Hasil Uji		
		Tampilan Aplikasi	Hasil Diharapkan	Kesimpulan
1.	Menu <i>Login</i>		Aplikasi menampilkan tampilan <i>login</i> yang nantinya akan digunakan oleh admin	[✓] Berhasil [] Tidak Berhasil
2.	<i>User</i>		Menampilkan Submenu yang berisi data <i>login</i> , Data <i>Role</i> , Data Pegawai, Data <i>Customer</i>	[✓] Berhasil [] Tidak Berhasil
3.	Master Data		Menampilkan Submenu yang berisi <i>engine</i> , <i>merk</i> , <i>chassis</i> , <i>vehicle</i> , varian, <i>Pengajuan</i> , Data Master.	[✓] Berhasil [] Tidak Berhasil
4.	Master Gambar		Menampilkan Submenu yang berisi Data Gambar	[✓] Berhasil [] Tidak Berhasil

Table 4. Hasil Pengujian Halaman *Drafter*

NO	Komponen yang diuji	Skenario dan Hasil Uji		
		Tampilan Aplikasi	Hasil Diharapkan	Kesimpulan
1.	Menu <i>Login</i>		Aplikasi menampilkan tampilan <i>login</i> yang nantinya akan digunakan oleh <i>drafter</i>	[✓] Berhasil [] Tidak Berhasil
2.	<i>WorkSpace</i>		Menampilkan data lembar kerja. Selain itu pengguna juga dapat menambah, edit dan <i>delete</i> data	[✓] Berhasil [] Tidak Berhasil

Apabila terdapat kesalahan atau *bug* pada saat pengujian pada program maka akan di kembalikan pada tahap pengkodean. Begitu juga jika ada kesalahan dalam tampilan maka akan di kembalikan kepada desain.

E. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

1. **Pemeliharaan Korektif**
Melakukan perbaikan *bug* atau kesalahan yang ditemukan setelah aplikasi dirilis.
 - Pelaporan *Bug*: Memastikan ada sistem pelaporan *bug* yang efektif sehingga pengguna dapat melaporkan masalah dengan mudah.
 - Tim Pemeliharaan: Menyediakan tim yang bertanggung jawab untuk mendiagnosis dan memperbaiki *bug* secara cepat.
 - Prioritas *Bug*: Mengelompokkan *bug* berdasarkan tingkat keparahan dan prioritas perbaikan.
2. **Pemeliharaan Adaptif**
Melakukan modifikasi aplikasi agar sesuai dengan perubahan lingkungan atau teknologi.
 - Pemantauan Perubahan Teknologi: Memantau perkembangan teknologi dan memastikan aplikasi tetap kompatibel dengan sistem operasi, perangkat keras, atau perangkat lunak lainnya.
 - Migrasi Data: Mengelola migrasi data jika ada perubahan dalam platform atau *database* yang digunakan.
3. **Pemeliharaan Perfektif**
Meningkatkan atau mengoptimalkan kinerja aplikasi berdasarkan umpan balik pengguna dan analisis performa.
 - Umpan Balik Pengguna: Mengumpulkan umpan balik dari pengguna untuk mengetahui area yang perlu ditingkatkan.
 - Optimasi Kode: Meninjau dan mengoptimalkan kode untuk meningkatkan efisiensi dan kinerja.
 - Pembaharuan Fitur: Menambahkan fitur baru atau meningkatkan fitur yang sudah ada untuk memenuhi kebutuhan pengguna yang berkembang.
4. **Pemeliharaan Preventif**
Melakukan antisipasi dan mencegah masalah yang mungkin terjadi di masa depan.
 - Audit Keamanan: Melakukan audit keamanan secara berkala untuk mengidentifikasi dan memperbaiki potensi kerentanan.
 - *Backup dan Recovery*: Menyediakan sistem backup yang handal dan rencana pemulihan bencana.
 - Pemantauan Sistem: Menggunakan alat pemantauan untuk mendeteksi dan menangani masalah sebelum mempengaruhi pengguna.
5. **Dokumentasi**
Menyediakan dokumentasi yang lengkap dan *up-to-date* untuk mendukung pemeliharaan dan dukungan aplikasi.
 - Dokumentasi Pengguna: Menyediakan panduan pengguna dan FAQ untuk membantu pengguna menyelesaikan masalah umum.
 - Dokumentasi Teknis: Menyediakan dokumentasi teknis yang mendetail untuk tim pemeliharaan, termasuk arsitektur sistem, diagram alur kerja, dan panduan pengembang.

4. Kesimpulan

Dengan adanya permasalahan-permasalahan tersebut maka dibuatkan aplikasi SIMAGA yaitu sebuah sistem informasi pengelolaan stok gambar yang dapat membantu antar pengguna dalam menyimpan dan mengelola stok gambar secara terpusat, terintegrasi, efisien dan dapat diakses oleh banyak pengguna. Dengan aplikasi SIMAGA yang diaplikasikan pada CV. Berkah Nusantara Jaya diharapkan dapat membantu meningkatkan kinerja dari perusahaan tersebut, karena :

1. Pengelolaan gambar desain menjadi terpusat dan otomatis terinput secara sistem
2. Pencarian gambar desain menjadi lebih mudah dan dapat mengefisiensi waktu saat pencariannya
3. Melacak riwayat perubahan gambar desain menjadi mudah karena *record* pekerjaan yang sudah otomatis terekam secara sistem
4. Data gambar desain yang tersimpan terpusat telah menghilangkan potensi kehilangan gambar desain yang sebelumnya dikhawatirkan

Referensi

- Abdilah, A., Nurlelah, E., Hasan, F. N., & Utami, D. Y. (n.d.). *This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License Aplikasi Sistem Informasi Koperasi Simpan Pinjam Berbasis Web Pada PT. Mitraindo Sejahtera Utama Tangerang*. <https://doi.org/10.31294/jtk.v4i2>
- Ardi Widodo, K., Adi Wibowo, S., & Vendyansyah, N. (2021). Penerapan Sequential Search Untuk Pengelolaan Data Barang. *Jurnal Ilmiah Teknik Informatika*, 15(1), 86–97.
- Aurin, K., Fajar, M., Munir, A., Stmik, I., & Makassar, K. (2021). Pemodelan Jaringan Sensor Nirkabel Menggunakan System Modeling Language. *JTRISTE*, 8(1), 9–20.

- Bahrudin, A., Jupriyadi, & Permata. (2020). Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus :Studio Muezzart). *Jurnal Ilmiah Infrastruktur Teknologi Informasi*, 1(2), 14–18.
- Danuri, M. (2019). *Perkembangan Dan Transformasi Teknologi Digital*.
- Fathun. (2020). *SERI OTOMOTIF SMK : Gambar Teknik Otomotif Untuk SMK/MAK Kelas X* (NILACAKRA, Ed.; 1st ed., Vol. 1). NILACAKRA.
- Rahmi, E. R., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik*, 7(1), 821–834.
- Supiyandi, S., Zen, M., Rizal, C., & Eka, M. (2022). Perancangan Sistem Informasi Desa Tomuan Holbung Menggunakan Metode Waterfall. *JURIKOM (Jurnal Riset Komputer)*, 9(2), 274–280. <https://doi.org/10.30865/jurikom.v9i2.3986>
- Wahid, A. A. (2020). Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK Oktober (2020) Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, 1–5.
- Yulia, Arizona, N. D., & Anisa, T. (2021). Jurnal Sistem Informasi Akuntansi Pendapatan Jasa Dan Penjualan Suku Cadang Pada Bengkel Podomoro Motor Pontianak Berbasis Web. *Jurnal Sistem Informasi Akuntansi*, 86–94.