I. PENDAHULUAN

1.1 Latar Belakang

Tanggung jawab Dinas Perhubungan Kota Bandar Lampung adalah mengkoordinasikan berbagai aspek sistem transportasi wilayah kota atau kebijakan transportasi. Dinas Perhubungan Kota Bandar Lampung yang sering disebut Dishub ini terletak di Jalan Zainal Abidin Pagar Alam, Lampung Selatan, Kecamatan Rajabasa Kota Bandar Lampung. Dishub berwenang menerbitkan berbagai izin terkait angkutan melalui kantor ini, antara lain: penerbitan surat izin angkutan penumpang, izin angkutan penumpang, izin angkutan barang, izin trayek dan kartu pengawasan angkutan penumpang umum, izin trayek angkutan antar jemput, izin penyelenggaraan angkutan sewa, izin penyelenggaraan angkutan wisata, surat izin trayek (SPIT), izin beroperasi (SPIO), angkutan taksi antar kota (AKDP), dll. (N.Thano, 2022).

Gedung *Area Traffic Control System* atau yang lebih dikenal dengan istilah (ATCS) didirikan pada tahun 2014, kemudian diresmikan pada tanggal 6 Januari 2015. ATCS memiliki 11 pegawai honorer dan 5 pegawai negeri sipil. Dengan Kepala Bidang Lalu Lintas Bapak Iskandar Z.ATD, SH, MT, Kepala Satuan Tugas dipimpin oleh Bapak Nirma Thano,S.Si, T,M.M, Kepala Unit dipimpin oleh Bapak Drs. Haidirsyah, Komandan Regu A oleh Bapak Sriyono, Komandan Regu B oleh Bapak Ferianto, dan Manajemen Rekayasa Lalu Lintas atau (MRLL) dipimpin oleh Bapak Ahmad Junaidi ,S.IP.

Salah satu jenis pengendalian lalu lintas yang ditawarkan oleh ATCS adalah pengelolaan pelanggaran. Selain itu, ATCS memiliki ruang kontrol yang dapat mengatur rekayasa lalu lintas melalui pengawasan CCTV di berbagai persimpangan jalan. *Traffic Engineering Management* (MRLL) adalah seperangkat perusahaan dan operasi yang mencakup perencanaan, pengadaan, pemasangan, pengaturan, dan pemeliharaan fasilitas peralatan jalan dalam rangka mewujudkan, mendukung, memelihara keamanan, keselamatan, ketertiban, dan kelancaran lalu lintas(N.Thano, 2022).

Absensi pada ATCS dan MRLL Dishub masih dilakukan dengan paraf, pada lembar absensi terdapat nama, pangkat/golongan, jabatan, dan tanggal absensi.

Absensi diserahkan pada bagian tata usaha yang akan diinputkan ke *Microsoft Office Excel*, sedangkan untuk laporan harian karyawan hanya menuliskan pada buku masing-masing karyawan. Maka, tujuan dibuatnya pengembangan sistem untuk mengoptimalkan sistem yang berjalan.

Solusi yang diperlukan berdasarkan permasalahan diatas adalah pengembangan sebuah aplikasi absensi dan laporan harian karyawan berbasis web yang dapat mengoptimalkan serta dapat membantu karyawan dalam mengelola absensi dengan menggunakan citra wajah yang nanti nya akan di verifikasi oleh kepala satuan tugas, lalu absensi akan diserahkan kepada staf tata usaha setiap bulan, laporan harian yang akan diinput setiap hari oleh karyawan sama seperti absensi nantinya akan diverifikasi oleh kepala satuan tugas untuk evaluasi karyawan dan penambahan penjadwalan yang dapat menata kegiatan para karyawan seperti jadwal rapat bulanan.

Aplikasi akan dibangun menggunakan metode *prototype,* pengembangan dengan metode *prototype* dipilih karena lebih cepat dan dapat lebih optimal. Aplikasi yang akan dibangun berjudul "Aplikasi Absensi Dan Laporan Harian Karyawan ATCS Dan MRLL Pada Dinas Perhubungan Kota Bandar Lampung".

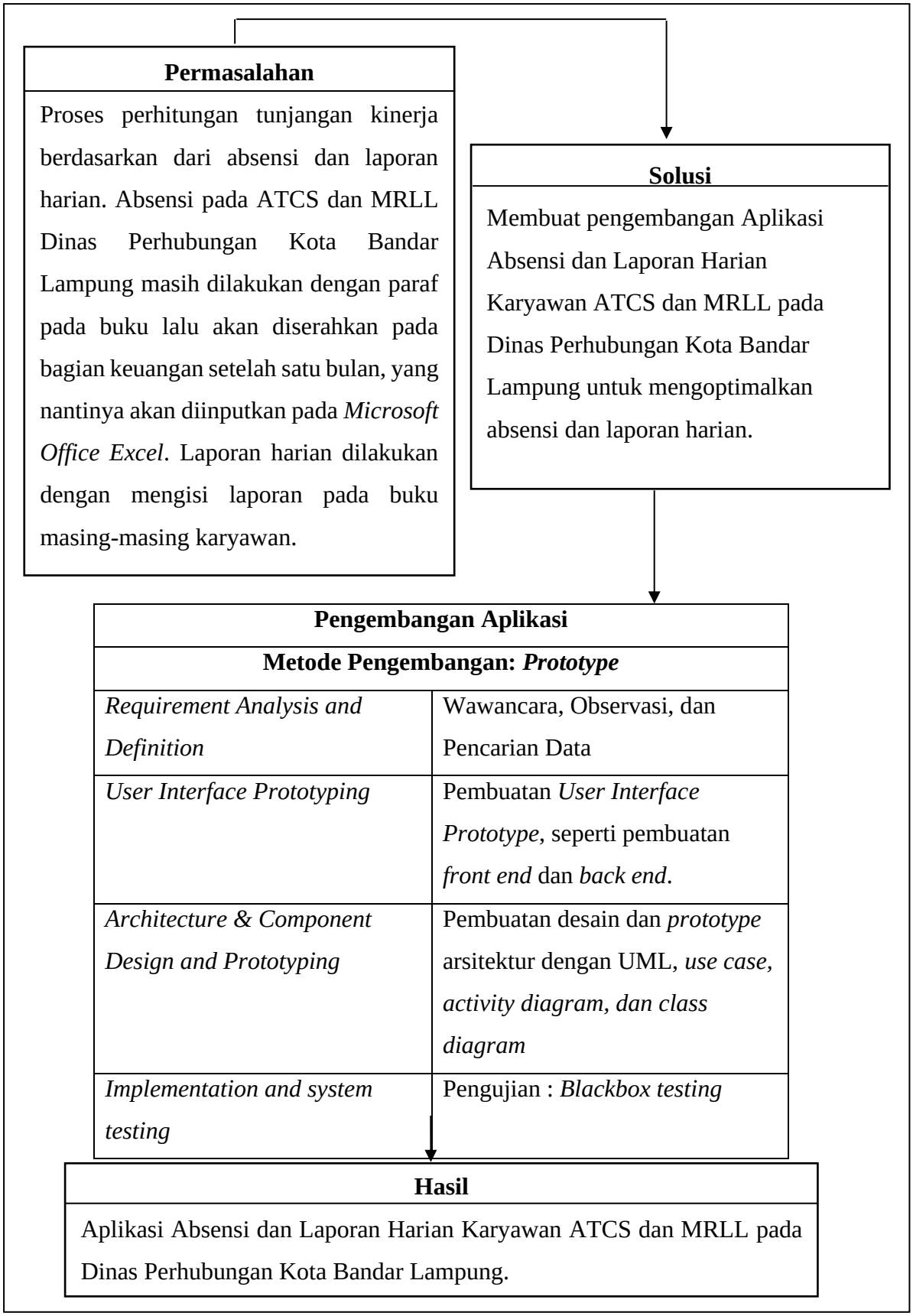
1.2 Tujuan

Tujuan yang sudah diraih pada penulisan tugas akhir ini yakni menghasilkan Aplikasi Absensi dan Laporan Harian Karyawan ATCS dan MRLL Pada Dinas Perhubungan Kota Bandar Lampung yang dapat mempermudah proses absensi, penjadwalan, dan laporan.

1.3 Kerangka Pemikiran

Proses absensi masih dilakukan dengan paraf lalu diserahkan pada Bagian Tata Usaha yang nantinya akan diinputkan ke *Microsoft Office Excel*, sedangkan untuk laporan harian menuliskan pada buku masing-masing karyawan, sehingga tidak adanya laporan harian yang tersampaikan.

Berdasarkan permasalahan diatas maka diperlukan pengembangan untuk membantu karyawan melakukan absensi dan laporan harian, yaitu dengan membuat "Aplikasi Absensi dan Laporan Harian Karyawan ATCS dan MRLL pada Dinas Perhubungan Kota Bandar Lampung". Metode yang digunakan yaitu metode *prototype*. Bagan kerangka pemikiran akan disajikan dalam Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Aplikasi ini memiliki kontribusi kepada beberapa pihak yang diuraikan sebagai berikut:

1. Dinas Perhubungan Kota Bandar Lampung
   a. Mempermudah membuat rekapan absensi bulanan.
   b. Mempermudah dan membantu rekapan laporan harian.
   c. Membantu penjadwalan kegiatan ATCS dan MRLL.
2. Politeknik Negeri Lampung
   a. Menyumbang karya tulis kepada Politeknik Negeri Lampung.
   b. Menambah wawasan dan pengetahuan kepada mahasiswa Politeknik Negeri Lampung.

II. TINJAUAN PUSTAKA

2.1 Aplikasi *Website* dan *Frontend*

Kata *application* berasal dari kata kerja *to apply* dan berarti "pengolah" dalam bahasa Indonesia. Perangkat lunak yang dibuat untuk membantu pengguna dalam melaksanakan tugas tertentu dan memanfaatkan program lain oleh aktor sasaran (Mahardika, 2020). Menurut (Bekti & Humaira, 2015), situs web adalah kumpulan halaman statis dan dinamis yang disusun dalam jaringan bangunan yang saling berhubungan dan mencakup informasi yang disimpan secara online dan dapat diakses dan dilihat pada perangkat yang dapat mengakses internet. *World Wide Web* (www) ialah sistem jaringan berbasis client-server dengan media HTTP (*Hypertext Transfer Protocol*) dan TCP (*Transmission Control Protocol*/*Internet Protocol*). Kata *web* ialah versi singkat dari www. Fungsi *website* secara umum yaitu alat komunikasi, media informasi, media entertainment, dan sarana transaksi. Alasan mengapa banyak perusahaan atau individu menggunakan *website* antara lain (Soejono, Setyanto, & Sofyan, 2018):

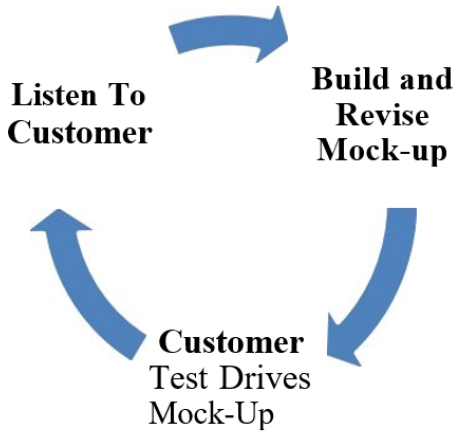
1. *Website* adalah sarana komunikasi pengenalan sebuah produk kepada masyarakat lebih luas, seperti visi misi perusahaan, dan jenis produk yang ditawarkan.
2. Internet merupakan media informasi tanpa batas, teknologi semakin maju segala sesuatu dilakukan memakai situs website.
3. *Website* didesain secara menarik, sehingga dapat meningkatkan citra bisnis perusahaan dan memudahkan pekerjaan.

Frontend merupakan suatu bagian dari sistem yang menyajikan tampilan kepada pengguna. *Frontend* bertanggung jawab pada tampilan antar muka pengguna dan bertugas meningkatkan bagian-bagian visible dari suatu sistem. *Frontend* adalah *developer* harus membuat *consumer interface* agar desain menarik perhatian dan tidak membosankan pada sisi pengguna. *Frontend* dibangun dengan menggunakan banyak komponen seperti HTML, CSS, dan *JavaScript* (Prasojo, 2021).

2.2 Metode *Prototype*

Metode pengembangan yang akan dimanfaatkan yakni *Prototyping*, tujuan menggunakan metode ini didasarkan pada konsep model kerja. Dengan *prototyping* sistem akan dikembangkan lebih cepat dibandingkan metode secara manual.

Pengertian metode *prototype* yaitu pengembangan perangkat lunak yang berfungsi sebagai langkah awal dari sistem. Metode *prototyping* melihat kebutuhan dan masukkan pengguna, pengembang sistem dan pengguna saling berkomunikasi satu sama lain mengenai penyamaan sistem yang mendasari pengembangan sistem operasional, pengguna juga dapat berpartisipasi dalam menentukan model, sistem yang dibangun juga mempunyai kualitas yang diharapkan sesuai dengan kebutuhan yang ada (Pradipta, 2019). Metode pengembangan sistem *prototyping* disajikan dalam Gambar 2 (Dieterici, 2018).



Gambar 2. *Prototyping Model*

Pada Gambar 2 terdapat tiga tahapan yang akan disajikan sebagai berikut (Dieterici, 2018):

1. *Listen to Customer* (Mendengarkan Pelanggan)

   Pada tahap ini pengembang bisa mendapatkan informasi tentang permasalah apa saja yang terjadi. Data yang didapat dari permasalahan tersebut akan disajikan sebagai pedoman untuk tahap pengembangan selanjutnya.

2. *Build and Revise Mock-up* (Membangun dan Memperbaiki *Prototype*)

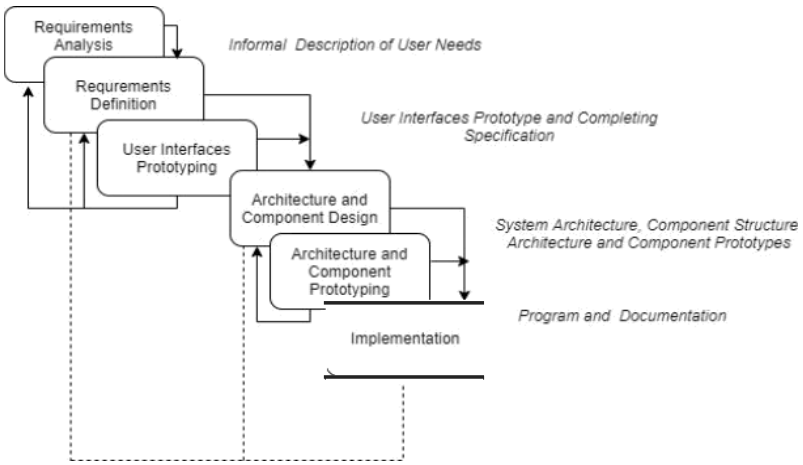
   Selanjutnya akan dilakukan proses perancangan *prototype* pada sistem jika kebutuhan sistem sudah terkumpul, tahapan-tahapan sebagai berikut:

- Perancangan proses di dalam sistem meliputi *input* (masukan), *output* (keluaran)
- Perancangan UML (*Unified Modelling Language*) hal ini dilakukan agar apa saja dibutuhkan dan bagaimana alur sistem tersebut dijalankan. Desain UML yang dimanfaatkan dalam sistem meliputi: *Use Case*, *Activity Diagram*, dan *Class Diagram*.
- Perancangan *Interface* (antarmuka) dan fitur yang diperlukan.

3. *Customer Test Drives Mock-Up* (Pengujian *Prototype*)

   Pada tahap ini akan dilaksanakan pengujian, tahapan *prototype* sistem yang sudah dirancang dan mengevaluasi apakah sistem *prototype* yang telah dibuat sesuai dan memenuhi harapan. Proses pengujian *prototype* sistem menggunakan pengujian *BlackBox Testing*.

Tahapan dan alur pengembangan sistem dengan teknik *Prototyping Oriented Software*, pada tahap pertama dilaksanakan analisis dan pendefinisian kebutuhan, tahap kedua mengidentifikasi kebutuhan dan pembuatan *front end* dan *back end*, tahap ketiga adalah pembuatan arsitektur dan desain menggunakan UML, tahap terakhir yaitu pengkodean dan pengujian atau testing.Disajikan pada Gambar 3 (Dieterici, 2018).



Gambar 3. Tahapan dan alur *prototyping*

2.3 Perancangan dan Desain

Perancangan merupakan langkah mendeskripsikan merencanakan, menggambarkan dan sketsa atau menyusun beberapa komponen *independen* sebagai satu fungsi yang utuh. Perancangan sistem bisa digambarkan dalam bentuk *mapping chart*, yaitu suatu alat grafis yang dapat menampilkan proses sistem secara runtut (Munawar, Raharjo, & Megawati, 2021).

Sedangkan desain merupakan proses mendeskripsikan, menggambarkan, merencanakan serta menyusun beberapa komponen *independen* menjadi satu kesatuan *fungsional* yang lengkap. Desain atau rancangan mempunyai dua tujuan, yaitu untuk memberikan ilustrasi yang jelas bagi *programmer* dan ahli lainya sebagai akibatnya bisa memenuhi kebutuhan pengguna sistem. Desain juga melibatkan eksperimen atau beberapa percobaan seperti sketsa dan mencoba beberapa konsep dan pandangan baru (Meliyana, 2017).

2.3.1 *Mapping Chart*

Mapping chart adalah representasi grafis dari langkah dan alur program. Mapping chart dapat digunakan untuk menggambarkan proses dan prosedur dalam suatu organisasi. Diagram pada gambar digunakan untuk menggambarkan urutan alur kerja/proses dalam pembuatan sistem (Widarma & Rahayu, 2017). Simbol kartu pemetaan dapat ditemukan pada Tabel 1:

Tabel 1. Simbol *Mapping Chart*

Simbol	**Keterangan**
(1)	**(2)**
	Terminal digunakan untuk menunjukan sumber atau dokumen dan laporan
	Menyatakan output yang digunakan
	Operasi manual
	Konektor satu intera halaman

	Garis alir dokumen
	File untuk menyimpan dokumen sumberdan laporan
	Catatan akuntansi (jurnal, register, log,buku besar)

Sumber: Hall (2019).

2.3.2 *Unified Modelling Language* (UML)

Bahasa standar untuk merancang perangkat lunak atau desain program disebut *Unified Modeling Language* (UML). Pada dasarnya, UML digunakan untuk merancang dan membuat dokumen artefak perangkat lunak. Saat ini, UML adalah bahasa yang diakui untuk membuat cetak biru perangkat lunak (Steven, 2019).

2.3.3 *Use Case* Diagram

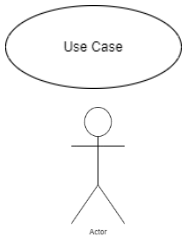
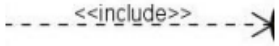
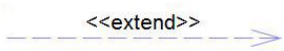
Hubungan antara satu atau lebih aktor dan sistem informasi masa depan digambarkan dalam diagram *use case*. *Use case* digunakan untuk mengidentifikasi fungsionalitas yang dapat digunakan dalam sistem informasi dan siapa yang berwenang untuk melakukannya (Kurniawan, 2020).

Ketentuan dalam penamaan *use case* diagram adalah bahwa nama didefinisikan dengan cara yang sederhana dan mudah dimengerti, dua hal dasar dalam pendefinisian *use case* yaitu (Shalahuddin & Rosa, 2015).

1. Aktor adalah orang, proses, atau sistem lain yang berada di dalam dan di luar sistem yang sedang dikembangkan. Simbol aktor adalah gambar manusia.
2. Use case adalah unit yang didefinisikan oleh sistem sebagai pertukaran pesan yang efektif antara unit atau aktor lain.

Simbol-simbol *use case* dapat dilihat pada Tabel 2.

Tabel 2. *Use case* diagram

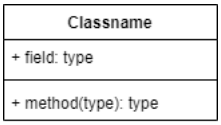
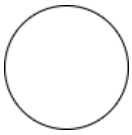
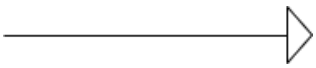
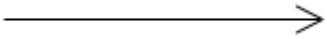
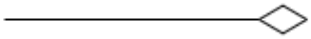
Simbol	Keterangan
(1)	(2)
	Berfungsi untuk menggambarkan interaksi antar sistem dan aktor Aktor mewakili individu, prosedur, atau sistem lain yang terlibat dalam *use case.*
	Untuk komunikasi antara aktor dan *use case,* asosiasi antara aktor dan *use case* ditampilkan sebagai garis tanpa panah.
	Asosiasi antara aktor dan kasus penggunaan yang secara pasif berinteraksi dengan sistem menggunakan panah terbuka.
	*Include*, hubungan antara *use case* tambahan dan *use case* di mana *use case* tambahan bergantung pada *use case* asli agar dapat berfungsi.
	*Extend*, ketika *use case* yang dimaksud dapat diakses secara langsung tanpa *login*.

Sumber: (Shalahuddin & Rosa, 2015)

2.3.4 *Class* Diagram

Jenis diagram yang paling umum adalah *class diagram*, yang digunakan untuk menunjukkan struktur kelas suatu sistem. *Class diagram* dalam model desain (tampilan logis) suatu sistem juga menampilkan hubungan antar kelas dan penjelasan menyeluruh dari setiap kelas. *Class diagram* berguna untuk menangkap struktur semua kelas yang membentuk arsitektur sistem yang dikembangkan selama fase desain. Menurut (Shalahuddin & Rosa, 2015), diagram kelas secara khusus mengandung istilah-istilah berikut: kelas (*class*), asosiasi (relasi), generalisasi (generalisasi), agregat (kumpulan), operasi atau prosedur (operation), atribut (atribut), dan visibilitas (visibilitas). Simbol-simbol *class diagram* dapat dilihat pada Tabel 3.

Tabel 3. *Class diagram*

Simbol	Keterangan
(1)	**(2)**
	*Class*, sekelompok item dengan sifat dan fungsi yang serupa.
	Antar muka / *interface*, dalam pemrograman berorientasi objek sama.
	Asosiasi / *association*, memerlukan multiplisitas serta asosiasi antar kelas.
	Generalisasi dan hubungan spesialisasi antar kelas (umum dan khusus).
	Hubungan antar kelas dengan semantik ketergantungan antar kelas disebut ketergantungan.
	Agregasi / *aggregation*, relasi kelas dengan pengertian *all – part.*

Sumber: (Shalahuddin & Rosa, 2015)

2.3.5 *Activity* Diagram

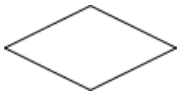
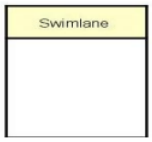
Activity diagram menunjukkan banyak aliran aktivitas pada sistem yang sedang dibuat, awal, keputusan potensial, dan akhir. *Activity* diagram dapat digunakan untuk menunjukkan proses konkuren yang mungkin terjadi selama beberapa eksekusi (Kurniawan, 2020). Berikut ini didefinisikan menggunakan *Activity* diagram (Shalahudin dan Rosa, 2013):

1. Setiap rangkaian kegiatan yang disebutkan memiliki proses bisnis sistem yang ditetapkan yang dibangun ke dalam desainnya
2. Setiap aktivitas memiliki desain antarmuka tampilan, dan antarmuka sistem diatur ke dalam kelompok (antarmuka pengguna).
3. Setiap kasus uji aktivitas harus memiliki pengujian yang disyaratkan oleh desain pengujian aktivitas.

4. Tata letak menu perangkat lunak terlihat

Simbol-simbol *activity diagram* dapat dilihat pada Tabel 4.

Tabel 4. *Activity* diagram

Simbol	**Keterangan**
(1)	**(2)**
	*Start point*, digunakan untuk menunjukan awal dari aktivitas.
	*End point*, digunakan untuk mengakhiri aktivitas.
	*Activities*, berfungsi untuk menunjukan interaksi antara antarmuka atau tampilan.
	Dekomposisi ditunjukkan dengan penggunaan join (penggabungan) atau rake.
	Poin keputusan adalah pilihan untuk membuat keputusan, seperti benar atau salah.
	Diagram swimlane yang menunjukkan pembagian kerja digunakan.

Sumber: (Shalahuddin & Rosa, 2015)

2.4 *BlackBox Testing*

Pengujian *BlackBox* didefinisikan sebagai pengujian perangkat lunak terhadap spesifikasi fungsional tanpa memverifikasi desain atau kode program (Wiradiputra, Candiasa, & Divayana, 2021), Tes ini mencari sejumlah kelemahan, termasuk masalah kinerja, fungsi yang rusak, masalah antarmuka, struktur data atau masalah basis data, dan kesalahan dalam kategori. Ada banyak cara untuk melakukan pengujian *blackbox*, termasuk pengujian perilaku, kinerja, dan persyaratan. Pengujian *blackbox* sering mengungkapkan hal berikut:

1. Fungsi yang tidak benar atau tidak ada,
2. Kesalahan *interface*,
3. (*Performance error*),
4. Kesalahan inisialisasi dan terminasi.