

BAB I. PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi berperan penting dalam kehidupan masyarakat, perkembangan teknologi informasi saat ini berjalan semakin pesat. Teknologi informasi memberikan kenyamanan, kecanggihan, kemudahan, dan keamanan pada data yang digunakan. Adapun tujuannya yaitu untuk mempermudah dan meningkatkan pekerjaan pada suatu Instansi baik pendidikan, pemerintahan maupun swasta. Banyak yang menerapkan teknologi informasi dalam setiap kegiatan, hampir semua kegiatan yang berlangsung tidak jauh dari teknologi informasi, salah satunya adalah pengelolaan kepegawaian. Umumnya instansi menggunakan sebuah sistem informasi dalam mengelola pegawainya. Sistem informasi manajemen merupakan suatu sistem yang memiliki beberapa komponen yang saling berkaitan untuk mencapai tujuan. Ada 3 kegiatan sistem informasi manajemen seperti *input* data, memproses data, dan menghasilkan informasi. Pada setiap kegiatan sangat membutuhkan informasi yang baik seperti perencanaan, pelaksanaan, pengorganisasian dan pengawasan. Informasi manajemen sangat dibutuhkan dalam pengambilan keputusan agar dapat diakses dengan mudah dan menghasilkan informasi yang akurat (Hutahaean dkk., 2021).

Pengelolaan arsip adalah suatu kegiatan yang ditujukan untuk pengelolaan seluruh dokumen yang ada. kegiatan yang menghasilkan informasi yang dibuat oleh instansi atau organisasi yang disimpan secara teratur. Salah satu arsip yang dikelola oleh sebuah instansi yaitu arsip data kepegawaian (Lestari dkk., 2019). Arsip mempunyai rekam peristiwa atau kegiatan yang berbentuk media, maka pengelolaan arsip pegawai penting dilakukan untuk sebuah instansi.

PT XYZ adalah salah satu perusahaan (BUMN) Badan Usaha Milik Negara yang memiliki tanggung jawab untuk membangkitkan menyediakan sumber listrik di Sumatera bagian selatan dan tengah. berdasarkan fungsi dibagi menjadi 3 bagian pembangkitan, transmisi, dan distribusi. Perusahaan ini memiliki beberapa divisi bagian yaitu : *Engjneering*, Ophar (operasi dan pemelihara), dan KSA (logistik, akuntansi, SDM dan umum). PT XYZ memiliki data kepegawaian, yang dikelola oleh bagian SDM yang dimana di dalamnya terdapat data masing-masing pegawai.

Mulai dari data pribadi, keluarga, kenaikan pangkat, mutasi, cuti dll.

Saat ini pengelolaan sistem manajemen data pegawai pada PT XYZ masih belum efektif dan efisien. Proses administrasi pegawai masih menggunakan cara yang manual, mulai dari pegawai diterima sampai dengan pensiun pemberkasan masih disimpan di dalam ordner map dan dikelompokkan berdasarkan tahun masuk pegawai. Ketika ada perubahan data, pencarian berkas pegawai memerlukan waktu yang tidak singkat.

Solusi yang dibutuhkan pada permasalahan di atas adalah dengan membuat sistem informasi manajemen pegawai yang berfokus pada proses cuti dan mutasi, sistem ini terdiri dari *frontend*, *backend*, dan *database*. pada penulisan tugas akhir ini penulis hanya memfokuskan pada desain *frontend* yang digunakan untuk menampilkan tampilan kepada pengguna agar sistem terlihat menarik. Sistem ini akan memudahkan admin dalam melakukan pengelolaan data pegawai melalui desain *layout* yang mudah dipahami oleh pengguna, serta akan di implementasikan layanan *web service* untuk mengembangkan aplikasi. *Web service* merupakan sistem pertukaran data berbasis JSON yang berinteraksi antar aplikasi melalui internet. *Software* yang akan dibangun menggunakan bahasa pemrograman *JavaScript*, *Library ReactJS* yang digunakan untuk membangun *front-end* pada web dan *template Chakra UI* digunakan untuk membuat kerangka kerja UI. Sistem ini akan mempermudah admin yang ada di PT XYZ dalam mencari data, memasukkan data, serta menghasilkan informasi dengan waktu yang singkat.

1.2 Tujuan

Tujuan penulisan Tugas Akhir ini adalah Perancangan *Front-End* Sistem Informasi Manajemen Pegawai Menggunakan Web Service Pada PT XYZ. Sistem aplikasi ini berfungsi untuk mendigitalisasi proses pengelolaan data dan informasi kepegawaian agar memudahkan administrasi kepegawaian, seperti, mutasi pegawai, cuti dan pensiun serta mempercepat admin dalam mengurus dan melengkapi berbagai berkas kepegawaian maupun syarat lain seperti SK pangkat, SK Jabatan SK mutasi, SK cuti dll.

1.3 Kerangka Pemikiran

Pengolahan sistem manajemen data pegawai pada PT XYZ saat ini belum efektif dan efisien. Penyimpanan data masih disimpan di dalam lemari dan belum tersimpan ke dalam *database* yang berisiko kehilangan data serta proses pencarian data masih membutuhkan waktu yang banyak. Permasalahan di atas membutuhkan solusi yang dapat digunakan untuk membantu Admin SDM dalam melakukan pencarian, penambahan, dan pembaruan data, yaitu dengan membuat “Perancangan *Front-end* Sistem Informasi Pegawai Menggunakan Web Service Pada PT XYZ”. Metode yang digunakan adalah RAD. Kerangka pemikiran pembuatan aplikasi sistem manajemen pegawai ini disajikan pada Gambar 1.

Permasalahan
<p>Proses administrasi pegawai masih dilakukan dengan cara manual yang membutuhkan waktu cukup lama dalam melakukan <i>input</i> data, memperbarui data, memproses data, dan menghasilkan informasi. Hal ini dapat memperlambat penyajian data yang dibutuhkan karena harus dilakukan di kantor pada hari kerja. Arsip kepegawaian masih disimpan dalam bentuk <i>hardcopy</i> serta belum dicadangkan ke dalam <i>database</i>, sehingga bisa mengakibatkan kerusakan dan kehilangan data.</p>
Solusi
<p>Membuat Perancangan <i>Front-End</i> Sistem Informasi Manajemen Pegawai Menggunakan <i>Web Service</i> pada PT XYZ, guna mempermudah Administrasi pada bagian SDM dalam mengelola data pegawai.</p>
Perancangan Sistem
<p>Metode pengembangan : <i>Rapid Application Development (RAD)</i></p> <ul style="list-style-type: none"> ➤ <i>Requirment Planning</i> : Pengumpulan data sebagai analisa ➤ <i>Design Workshop</i> : Membuat desain menggunakan <i>mapping chart</i>, <i>DFD</i>, <i>Flowchart</i>, <i>desain interface</i>. ➤ <i>Implementation</i> : Melakukan penulisan kode program, pengujian, dan perbaikan program apabila terdapat kesalahan ➤ <i>Cutover</i> : Penerapan sistem yang telah dibuat serta melakukan pengujian.
Hasil
<p><i>Front End</i> Sistem Informasi Manajemen Pegawai Menggunakan <i>Web Service</i> pada PT XYZ, untuk mempermudah Admin pada bagian SDM dalam mengelola data pegawai.</p>

Gambar 1. Kerangka pemikiran

1.4 Kontribusi

Kontribusi Perancangan *Front-end* Sistem Informasi Pegawai Menggunakan *Web Service* Pada PT. XYZ yaitu :

1. Kontribusi bagi perusahaan
 - a. Mempermudah admin dalam mengelola penyimpanan, pembaruan, penambahan data.
 - b. Memudahkan pegawai dalam mendapatkan data dan informasi kepegawaian secara cepat.
 - c. Mempermudah manajer dalam mengontrol data pegawai.
2. Kontribusi bagi Politeknik Negeri Lampung
 - a. Memberikan pengetahuan dan pengembangan ilmu
 - b. Menjadikan referensi bagi para mahasiswa yang akan atau sedang melakukan tugas akhir dengan tema sejenis.

BAB II. TINJAUAN PUSTAKA

2.1 Aplikasi Website dan Front-End

Aplikasi adalah perangkat komputer yang dapat dipakai sesuai dengan kebutuhan *user* dan mempunyai fungsi khusus sesuai dengan kemampuannya. Aplikasi juga digunakan untuk membantu dalam hal penerapan, menyimpan data, mempermudah pekerjaan, permasalahan. (Siregar & Melani, 2019). *Website* merupakan kumpulan informasi yang diakses melewati internet. Semua orang bisa menggunakannya selama terhubung pada internet. *Website* adalah halaman yang dikumpulkan dan dirangkum pada sebuah domain yang terdapat *World Wide Web* (WWW) di internet. *Website* digunakan untuk membuat halaman yang dapat berisi data, *text*, suara, gambar dan lainnya yang bisa diakses secara *online* (Romadhon dkk., 2021). Ada beberapa hal yang menjadikan perusahaan atau individu membuat *website* adalah :

1. *Website* adalah wadah yang bisa mempresentasikan produk dan instansi kepada publik seperti latar belakang, visi misi perusahaan, serta jenis produk yang akan ditawarkan.
2. *Internet* adalah media informasi yang dapat dilakukan tanpa mengetahui batas waktu dan tersedia dimana saja.
3. *Memiliki* tampilan yang didesain dengan rapi sehingga meningkatkan citra bisnis perusahaan.

Front-end merupakan salah satu bagian dari *website* dan bertugas untuk menampilkan tampilan kepada para pengguna, serta menghubungkan antara *user* dengan sistem *Backend*. *Front-end developer* lebih fokus ke bagian *interface*, desain grafis dan memiliki tanggung jawab agar setiap pengguna dapat memperoleh informasi dalam aplikasi agar terlihat menarik. Biasanya bahasa pemrograman HTML, CSS, dan *JavaScript* digunakan untuk pengembangan *Front-end* (Arhandi, 2016).

2.1.1 *JavaScript Object Notation (JSON)*

JavaScript Object Notation (JSON) dibuat berdasarkan bahasa pemrograman *JavaScript*. JSON merupakan sebuah format yang digunakan untuk berbagi data antara *server* dan *client*. Format ini dapat diterapkan pada bahasa pemrograman lainya seperti *Ruby*, *Python*, PHP, Java. Struktur JSON ditulis dengan format *key* (kata kunci) yang diikuti dengan *value* (nilai) dan mendukung beberapa tipe data seperti *array*, *object*, *string*, *number*, *boolean*, *null* (Hasanuddin dkk., 2022) . JSON mempunyai dua struktur (Firdaus dkk., 2019) , sebagai berikut:

1. Pasangan *name* dan *value* sebagai *object*, *struct*, *record*, *hashtable*, *dictionary*, *keyed-list*.
2. Daftar nilai (*anorderedlist of values*) biasanya dinyatakan sebagai *vector*, *array*, *list* atau *sequence*.

2.1.2 *JavaScript*

Bahasa pemrograman web yang mempunyai karakteristik *client side programming language*, *client side programming*. proses yang dilakukan oleh *client* disebut dengan *JavaScript*. *Client* ini terkait dengan browser seperti *Opera Mini*, *Mozilla Firefox*, dan *Google Chrome*. *JavaScript* tidak sama dengan bahasa pemrograman *Java*, penulisan *JavaScript* dapat disematkan pada dokumen HTML atau diubah menjadi dokumen terpisah, kemudian ditautkan ke dokumen yang sedang di proses. *JavaScript* menggunakan fungsi yang di dirancang untuk mengontrol yang dapat diakses oleh pengguna (Pahlevi dkk., 2018).

2.1.3 *ReactJS*

ReactJS adalah *library JavaScript* digunakan pada kerangka kerja *open-source* untuk membuat *user interface* yang dikembangkan oleh perusahaan IT ternama di dunia seperti *Facebook* dan *Instagram*. *Library* ini digunakan untuk membuat bagian *frontend* dari sebuah aplikasi. Keunggulan dari *ReactJS* dapat memberikan kecepatan, *scability*, dan *simplicity*. Dalam MVC (*model view control*) *ReactJS* hanya menampilkan bagian *view* (Panjaitan & Pakpahan, 2021).

React mempunyai tujuan yaitu untuk memberikan kesederhanaan, kecepatan, dan skalabilitas. *React* juga memiliki fitur yang paling digunakan yaitu JSX, komponen *stateful*, model objek dokumen virtual. *ReactJS* dapat membuat komponen yang terenkapsulasi sehingga dapat mengatur *state* nya sendiri dan menggabungkan komponen tersebut untuk membentuk *user interface* yang lebih kompleks. Hal ini dapat mempermudah developer dalam menangani hal yang tidak diinginkan. JSX adalah *sintaks* seperti HTML atau XML digunakan oleh *React* untuk memperluas *ECMAScript* sehingga teks seperti XML bisa digunakan dengan kode *JavaScript*. Tujuan JSX digunakan bersama dengan *React* adalah untuk mendeskripsikan tampilan UI sehingga *React* untuk menampilkan pesan yang berguna untuk developer (Nasution & Iswari, 2021).

2.2 *Web Service*

Sistem pertukaran data berbasis JSON atau XML yang memfasilitasi interaksi antar aplikasi melalui internet disebut dengan *web service*. Metode ini memungkinkan aplikasi *web service* di terapkan pada platform yang berbeda sehingga dapat berkomunikasi satu sama lain. *Web service* dapat dikembangkan dengan cepat karena *web service* dapat digunakan dimana saja. *Web service* tertuju pada aplikasi yang dikembangkan dengan konsep tertentu, yaitu standar pemrograman komunikasi berdasarkan internet *protocol* dengan tujuan aplikasi memanggil fungsi dan pertukaran data (Nugraha dkk., 2021). *Web service* juga memiliki kelebihan (Khairullah & Erwadi, 2021) seperti:

1. Lintas *platform* yang memungkinkan aplikasi yang berbeda bisa saling bertukar data.
2. Keamanan data yang lebih terjamin.
3. Sebagai penghubung antara aplikasi dengan *database*.
4. Memudahkan pertukaran data yang dilakukan dari jarak jauh.

Menurut (Choirudin & Adil, 2019) *web service* memiliki tiga komponen standar sebagai berikut :

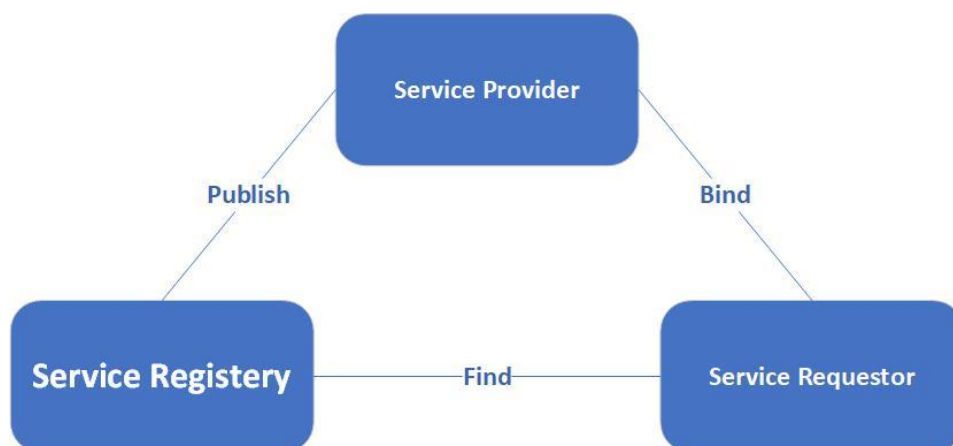
1. SOAP (*Simple Object Access Protocol*), protokol yang berhubungan dengan bertukar informasi pada jaringan terdistribusi.
2. WSDL (*Web Service Definition Language*), mengartikan keberadaan web dari web service dan dokumen standar yang ditulis dalam format XML.
3. UDDI (*Universal Description, Discovery, and Integration*), berbasis XML yang digunakan oleh suatu jaringan serta menunjukkan lokasi folder bebas platform yang berisi *service*.

2.2.1 Arsitektur Web Service

Menurut Laipaka (2018) arsitektur *web service* yang mendasari teknologi adalah *Service Oriented Architecture* (SOA), SOA mempunyai tiga entitas berbeda yang mempunyai tugas dari komponen *system*, yaitu :

1. *Service provider* merupakan suatu entitas yang mempersiapkan *web service* pada sistem untuk menjalankan tugas tertentu yang dapat digunakan oleh pengguna.
2. *Service requestor* merupakan entitas yang bekerja sebagai pengguna untuk membuat permintaan layanan ke *service provider*.
3. *Service registry* merupakan suatu entitas yang bekerja sebagai penyimpan (*repository*) suatu *software service* yang di publik oleh *service provider*.

Arsitektur *web service* disajikan pada Gambar 2.



Gambar 2. Arsitektur *web service*

Arsitektur *web service* mempunyai tiga operasi, seperti :

1. *Publish* bertugas untuk mempublikasikan layanan.
2. *Find* merupakan *service requestor* untuk mencari layanan yang dibutuhkan.
3. *Bind* merupakan *service requestor* setelah menemukan layanan yang berinteraksi dengan *service provider* serta mengakses layanan yang disediakan oleh *service provider*.

2.2.2 API

Application Programming Interface (API) adalah *interface* yang menggunakan akses layanan atau aplikasi dari program. API *mengizinkan* developer menggunakan fungsi yang telah tersedia pada aplikasi lain sehingga tidak harus membuat dari awal. API juga dapat memanggil fungsi melalui HTTP serta mendapatkan respons berupa XML atau JSON. Tujuan API adalah untuk berbagi informasi antar aplikasi yang berbeda dan mempercepat proses pengembangan aplikasi dengan menyediakan fungsi secara terpisah sehingga developer tidak harus merancang fungsi yang sama. API mendukung fungsi *Create, Read, Update, Delete* (CRUD) yang berjalan melalui *method* dan tipe *request* untuk melakukan komunikasi data seperti GET, POST, PUT, dan DELETE (Hasanuddin dkk., 2022).

2.2.3 REST

Salah satu *web service* yang mengimplementasikan konsep perpindahan antar *state* adalah konsep REST (*Representational State Transfer*). REST digunakan untuk *web service* yang berhubungan dengan *resource* (Sibagariang, 2016). REST merupakan suatu arsitektur komunikasi pada pengembangan layanan berbasis web yang biasanya bekerja dengan menggunakan protokol HTTP dengan *file* JSON atau XML sebagai sarana pertukaran data pada laman web. REST berinteraksi antara *client* dan *server* yang disediakan melalui *Universal Resource Identifiers* (URIs) dan *Method*. Masing – masing *Method* GET, POST, PUT, *DELETE* mempunyai fungsi tersendiri (Firdaus dkk., 2019a)

1. GET dipakai untuk membaca (*read*) suatu *resource*
2. POST dipakai untuk membuat (*update*) *resource*
3. PUT dipakai untuk membuat (*create*) suatu *resource*
4. DELETE dipakai untuk menghapus (*delete*) suatu *resource*

2.3 Sistem Informasi Manajemen Pegawai

Sistem yang dapat mengolah informasi atau data dan dukungan yang berguna untuk suatu organisasi. Sistem merupakan bagian atau elemen dari suatu himpunan yang saling berkaitan secara beraturan. Sistem didefinisikan sebagai pendekatan komponen dan prosedur. Sistem informasi ini bekerja secara terintegrasi dan memerlukan kelancaran pada sistem yang akan digunakan seperti *software* dan *hardware*. Pada sistem manajemen Informasi terdapat penyimpanan seperti *database* dan prosedur penerapan sistem manajemen perusahaan yang dijalankan oleh petugas untuk mengoperasikan sistem tersebut. (Hariyanto, 2016). Adapun tujuan dari sistem informasi manajemen yaitu :

1. Memberikan informasi agar dapat mengambil suatu keputusan dengan benar.
2. Memberikan informasi yang digunakan untuk perencanaan, evaluasi, serta perbaikan.
3. Memberikan informasi untuk menghitung jasa, produk dan tujuan lain.

Sistem informasi manajemen memiliki manfaat (Hutahaean dkk., 2021), yaitu :

1. Untuk memahami peluang ekonomis sistem informasi yang mengimplementasikan teknologi informasi baru.
2. Membantu memastikan kualitas dan keterampilan dalam menggunakan sistem informasi manajemen.
3. Mempermudah dalam mengakses data yang disediakan secara akurat dan tepat waktu untuk pengguna.
4. Mengembangkan perencanaan agar lebih efektif.
5. dapat mengidentifikasi kebutuhan untuk mendukung sistem informasi.

Meningkatkan pengembangan manajemen sistem serta pengolahan transaksi untuk meminimalisir pengeluaran dan memaksimalkan keuntungan

2.4 Perancangan dan Desain Sistem



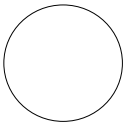
Perancangan adalah proses mendefinisikan dengan sebuah teknik yang berbeda, termasuk detail komponen dan deskripsi arsitektur, serta kendala yang dihadapi selama proses tersebut. Untuk menanggapi kebutuhan pengguna dan memberikan tampilan yang jelas dan lengkap merupakan tujuan perancang sistem (Aziz, N., Pribadi, G., & Nurcahya, 2020). Perancangan bisa digambarkan dalam bentuk diagram alir yang menampilkan proses secara urut.

Desain sistem merupakan sekumpulan komponen yang saling berhubungan kemudian diklasifikasikan dengan standarisasi yang jelas dan terintegrasi pada suatu sistem. Desain sistem bisa digunakan secara berulang oleh pengembang dan di sesuaikan dengan kebutuhan organisasi. Pada saat merancang sistem diperlukan pola pikir gambaran besar yang diperlukan untuk mengambil keputusan (Kurniawan & Fadlia Adiwijaya, 2021)




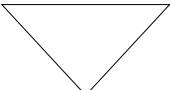

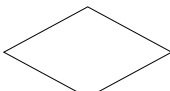

2.4.1 Mapping Chart

Mapping chart merupakan diagram alir yang mempunyai proses untuk menampilkan langkah dalam bentuk simbol-simbol grafis. *Mapping chart* membantu dalam menganalisis penelitian untuk memecahkan suatu masalah. Fungsi *mapping chart* adalah untuk menyederhanakan dan menggambarkan proses agar lebih mudah dilihat dan dipahami (Mukodimah & Muslihudin, 2019). Simbol *mapping chart* disajikan pada Tabel 1.

Tabel 1. Simbol-simbol *Mapping Chart*

Simbol	Nama	Fungsi
(1)	(2)	(2)
	Dokumen	Menyatakan penginputan dokumen yang dicetak
	Dokumen Rangkap	Menyatakan dokumen yang lebih dari satu
	<i>Connector</i>	Menyatakan sambungan satu dengan yang lainnya pada proses berikutnya

Tabel 1. Lanjutan simbol-simbol *Mapping Chart*


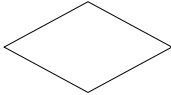

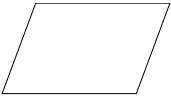



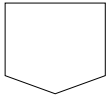
Simbol	Nama	Fungsi
(1)	(2)	(2)
	Proses	Proses pengolahan yang ditugaskan oleh komputer
	Proses Manual	Proses pengolahan yang dilakukan secara manual
	<i>Display</i>	Menyatakan <i>output</i> yang digunakan
	Arsip	Mengarsipkan data di dalam program
	Terminal	Awalan dan akhiran pada program
	<i>Decision</i>	Menyatakan kondisi yang menentukan 2 kemungkinan (iya atau tidak)
	<i>Connecting Line</i>	Menghubungkan simbol dengan simbol lainnya dengan menyatakan suatu alur proses

Sumber : (Fitriyana & Susianto, 2018)

2.4.2 *Flowchart*

Flowchart didefinisikan sebagai penyelesaian pada suatu masalah dengan menggambarkan langkah-langkah sebagai simbol-simbol. *Flowchart* juga merupakan penggambaran diagram alir yang menunjukkan alur di dalam sebuah program *flowchart* dapat mempermudah menyelesaikan suatu masalah terutama pada masalah yang perlu ditinjau dan dievaluasi lebih lanjut (Khesya, 2021). Simbol *mapping chart* disajikan pada Tabel 2.

Tabel 2. Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
(1)	(2)	(2)
	<i>Procces</i>	Proses Perhitungan/proses pengolahan data
	<i>Decision</i>	Menyatakan kondisi yang menentukan 2 kemungkinan seperti iya atau tidak.
	<i>Predefined Process</i> (Sub Program)	sub program
	<i>Input/Output Data</i>	Menghasilkan masukan/keluaran pada data
	<i>Terminator</i>	Permulaan/akhir program
	<i>On Page Connector</i>	Penghubung pada bagian <i>flowchart</i>
	<i>Flow Line</i>	Arah aliran program
	<i>Off Page Connector</i>	Penghubung pada bagian <i>flowchart</i> yang letaknya di halaman yang berbeda

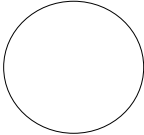




Sumber : (Fitriyana & Susianto, 2018)

2.4.3 DFD

Data *Flow* Diagram (DFD) adalah diagram yang menggambarkan proses entitas pada suatu sistem. Teknik grafis yang menggambarkan alir data *input* atau masukan dari *output* dapat diartikan sebagai DFD (Santoso & Nuralina, 2017). DFD juga merupakan suatu model logika yang memiliki proses untuk mengidentifikasi asal data, tujuan data, penyimpanan data, menghasilkan data. DFD menunjukkan gambaran yang berisi masukan, proses, dan keluaran dari suatu sistem.

DFD digambarkan dengan menggunakan tanda panah yang memiliki label, serta transformasi yang di tunjukkan menggunakan lingkaran-lingkaran (Soufitri, 2019). Simbol *mapping chart* disajikan pada Tabel 3.

Tabel 3. Simbol-simbol DFD

Simbol	Nama	Fungsi
	Proses	Notasi yang melakukan atau mempergunakan transformasi data.
	Entitas Eksternal	Berfungsi pada entitas dalam relasi di luar sistem
	Aliran Data	Aliran data dari sumber ke tujuan
		
	Penyimpanan	Penyimpanan data yang dikirim antar proses

Sumber : (Santoso & Nurmalina, 2017)

2.5 Metode Pengembangan *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah metode pengembangan yang termasuk dalam teknik bertingkat. Metode ini mengutamakan siklus pembangunan singkat, cepat, dan pendek. RAD menggunakan metode berulang dalam pengembangannya dan bertujuan untuk menetapkan kebutuhan pengguna (Prandawa & Muliawati, 2020). Metode pengembangan RAD disajikan pada Gambar 3.



Gambar 3. Metode pengembangan RAD

Sumber : (Putri & Effendi, 2018)

Rapid Application Development (RAD) mempunyai empat tahap yaitu :

1. *Requirment Planning*

Pada tahapan *requirment planning* akan dilakukan dengan cara mengumpulkan data yang akan dijadikan bahan analisa, serta mengidentifikasi tujuan sistem yang dibuat untuk mencapai tujuan yang telah dirancang.

2. *Design Workshop*

Tahapan *design workshop* akan melakukan perancangan desain pada perangkat lunak, dari hasil yang sudah di analisa pada tahap *Requirment Planning*. Rancangan yang dibuat berbentuk *Mapping Chart*, *DFD*, dan *Flowchart*.

3. *Implementation*

Tahapan *implementation* dimulai dari membuat program yang sudah dirancang pada tahap *Design Workshop*, dengan melakukan *coding* program, pengujian pada sistem, dan perbaikan jika terdapat kesalahan pada sistem.

4. *Cutover*

Tahapan ini dilakukan dengan penerapan pada sistem yang telah dibuat, kemudian dilanjutkan dengan proses pengujian. Tujuan dari tahapan ini adalah untuk memastikan sistem apakah terdapat kesalahan. Jika terdapat kesalahan pada sistem maka akan diperbaiki, agar sistem bisa digunakan dengan baik oleh pengguna.

2.6 Pengujian *System Usability Scale*

Pengujian menjadi alat untuk mengukur kualitas perangkat lunak yang dibutuhkan sebelum di publikasikan. Hal tersebut dilakukan untuk meyakinkan pengguna bahwa perangkat lunak digunakan untuk memenuhi kinerja sesuai dengan kebutuhan. Pada tugas akhir ini penulis menggunakan pengujian SUS. SUS adalah metode pengujian yang digunakan untuk memperoleh hasil dari pertimbangan jumlah sampel dan kepuasan penguji dalam menyelesaikan tugas serta melihat hasil responden.. Hasil perhitungan di ubah menjadi nilai yang menjadi dasar penentuan apakah aplikasi layak digunakan atau tidak (Nuriman dkk., 2019). Kelebihan dari *System Usability Scale* (SUS) adalah :

1. SUS menggunakan perhitungan yang mudah digunakan.
2. SUS bernilai 0-100.
3. Terbukti akurat meskipun menggunakan sampel yang kecil.
4. Disediakan gratis tanpa ada biaya.

2.7 Jurnal Terkait

Berikut ini beberapa jurnal yang menjadi bahan referensi pembuatan sistem.

1. Prawastiyo & Hermawan, (2020). Dengan judul “Pengembangan *Front-End Website* Perpustakaan Politeknik Negeri Jakarta Dengan Menggunakan Metode UCD (*User Centered Design*)”. Karya ilmiah ini dibuat untuk mempermudah dosen, mahasiswa, dan kalangan umum dalam mencari informasi. Pada aplikasi ini terdapat fitur tambahan yaitu fitur *upload*, fitur pencarian, fitur *download*, dan fitur plagiarisme. fitur ini bertujuan untuk mengelompokkan data agar lebih terstruktur. Metode pengembangan yang digunakan adalah metode *User Centered Design* (UCD)
2. Supriyanta dkk., (2019). Dengan judul “Aplikasi Sistem Manajemen Karyawan Pada PT. Gagas Mitra Jaya Yogyakarta”. Aplikasi ini dibuat karena sistem manajemen pegawai masih dilakukan secara manual pada sistem absensi, pengajuan cuti atau izin serta penukaran klaim nota bahan bakar. Sehingga dapat terjadi kesalahan dalam pencatatan dan memperlambat proses pencarian data yang dibutuhkan. Hasil dari penelitian ini adalah dengan dibuatnya aplikasi sistem manajemen pegawai untuk meminimalisir terjadinya kesalahan, mempercepat proses pengajuan cuti atau izin. Metode

pengembangan yang digunakan yaitu metode *Waterfall*.

3. Kuku, (2022). Dengan judul “Implementasi Perancangan Desain *Front-end* Pada Aplikasi Monitoring Pemasangan Jaringan Menggunakan Arsitektur *Microservice* Pada PT Lintas Data Multimedia”. Aplikasi ini dibuat karena proses pelaporan pemasangan jaringan oleh teknisi belum tersimpan ke dalam *database*, pengecekan pada pemasangan jaringan masih dilakukan dengan cara mendatangi langsung ke lokasi, sehingga dapat terjadi kehilangan data yang dapat mempersulit supervisor dalam menyiapkan laporan pemasangan jaringan. Hasil dari penelitian ini adalah dengan dibuatnya aplikasi *monitoring* pemasangan jaringan. Metode pengembangan yang digunakan yaitu metode RAD.
4. Aldo dkk., (2021). Dengan judul “Aplikasi Pelayanan Pada Klinik Kemina Dental Care Berbasis React.js dan *Database* NoSQL”. Aplikasi dibuat karena dalam melakukan penyimpanan data pasien maupun pengolahan data administrasi dilakukan secara manual, saat mencari *medical record* dan data pasien masih menggunakan buku pencatatan membutuhkan waktu yang lama. Hasil dari penelitian ini adalah dengan dibuatnya sistem informasi pelayanan pasien untuk mempermudah pengelolaan data dan memudahkan pelayanan pasien dalam laporan pasien, rekam medis, dan jadwal pemeriksaan. Aplikasi ini dibuat menggunakan *Library ReactJS* sebagai *front-end*.
5. Firdaus dkk., (2019). Dengan judul “Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer Polstri”. Aplikasi ini dibuat karena belum tersedianya aplikasi khusus yang terhubung ke internet untuk mempermudah mahasiswa dan dosen dalam mengakses perpustakaan *online*, sehingga tidak bisa dilakukan dari jarak jauh. Hasil dari penelitian ini adalah menjadikan sistem informasi yang *flexible*, *funksional*, serta dapat berintegrasi dengan aplikasi lainnya yang membutuhkan data dari perpustakaan.