

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian mengenai rekrutmen karyawan telah dilakukan oleh beberapa peneliti. Berikut merupakan beberapa penelitian yang dapat mendukung penelitian ini beserta dengan persamaannya dan perbedaannya.

Ardelia Astriany Rizky dan Irfan Ramdhani melakukan penelitian tentang Perancangan Sistem Informasi Perekrutan Karyawan Berbasis Web Menggunakan PHP Dan MYSQL Di PT. Ria Indah Mandiri. Sistem Informasi yang dibangun memiliki tujuan untuk membuat proses perekrutan karyawan menjadi terkomputerisasi dan menyelesaikan masalah penumpukan arsip lamaran kerja. Penelitian ini menggunakan metode pengembangan sistem *waterfall* dengan metode pendekatan berorientasi objek. [4]

Persamaan penelitian tersebut dengan yang dilakukan yaitu sama melakukan analisis terkait rekrutmen atau perekrutan karyawan dengan metode pendekatan berorientasi objek. Terdapat perbedaan yaitu pada penelitian sebelumnya hanya membahas lowongan kerja dan proses melamar yang tidak disertai seleksi. Sedangkan penelitian yang penulis lakukan yaitu terdapat kebutuhan karyawan, seleksi karyawan, pelatihan karyawan, metode pengembangan sistem *prototype*, dan penerapan arsitektur *microservice*.

Selanjutnya Ludy Septian Maulana, Novrini Hasti, dan Lasti Yossi Hastini melakukan penelitian Analisis Dan Perancangan Sistem Informasi Pengelolaan Karyawan Di CV. Kabita Informatika. Sistem informasi pada penelitian tersebut dibangun dengan tujuan untuk mempermudah dalam hal pengelolaan karyawan yang meliputi perekrutan karyawan dan penggajian. Penelitian dilakukan dengan menggunakan metode pendekatan berorientasi objek dan metode pengembangan sistem dengan metode *prototype*. Sistem informasi dibangun dengan bahasa pemrograman PHP dan *database* MySQL. [5]

Persamaan dari penelitian tersebut dengan penelitian yang dilakukan yaitu memiliki kesamaan melakukan rekrutmen karyawan, pengelolaan karyawan, seleksi lamaran, menggunakan metode pendekatan berorientasi objek, menggunakan metode pengembangan dengan *prototype*. Adapun perbedaan yaitu penelitian yang penulis lakukan hanya membahas mengenai kebutuhan karyawan, proses melamar, rekrutmen karyawan, dan pelatihan karyawan. Sedangkan penelitian sebelumnya membahas mengenai penggajian. Selain itu terdapat perbedaan lainnya yaitu penelitian sebelumnya masih menggunakan arsitektur monolitik sedangkan penelitian yang penulis lakukan menggunakan arsitektur *microservice*.

2.2. Sistem

Menurut (Jackson, 2010), Sistem adalah kumpulan dari unsur – unsur maupun komponen yang saling berhubungan dan bekerja sama guna mencapai tujuan yang sama. Sedangkan menurut (Jogianto, 2005), Sistem adalah sekelompok dari komponen

dan elemen yang saling dihubungkan untuk mencapai tujuan tertentu [6, p.3]. Berdasarkan definisi di atas maka dapat diambil kesimpulan. Sistem adalah suatu kumpulan yang terdiri dari unsur-unsur atau bagian-bagian yang saling terikat dan berhubungan sekaligus bekerjasama untuk mencapai tujuan tertentu.

2.2.1 Karakteristik Sistem

Model sebuah sistem terdiri dari *input*, proses, dan *output*. Suatu hal dapat dikatakan sebagai sistem apabila memiliki ciri tertentu. Berikut merupakan karakteristik sistem, yaitu : [7, p.13]

1. Komponen Sistem

Sistem terdiri dari banyak komponen yang saling berinteraksi guna membentuk kesatuan. Komponen dapat berupa suatu subsistem yang memiliki sifat menjalankan suatu fungsi yang mempengaruhi sistem utama.

2. Batasan Sistem

Batasan sistem berupa ruang lingkup yang dapat membatasi antara satu sistem dengan sistem lainnya.

3. Lingkup Luar Sistem

Hal apapun diluar batasan sistem yang dapat memberi pengaruh pada operasi sistem. Lingkungan luar dapat memberikan keuntungan dan memberikan kerugian pada sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem, sedangkan lingkungan luar

yang merugikan harus dapat dikendalikan karena dapat memberi pengaruh buruk pada kelangsungan hidup sistem.

4. Penghubung Sistem

Media yang menghubungkan satu sistem dengan yang lain disebut dengan *interface* atau penghubung sistem. Penghubung ini memungkinkan sumberdaya mengalir dan dapat dihubungkan satu sub sistem dengan yang lain. *Output* dari suatu subsistem dapat menjadi *input* bagi subsistem lain dengan menggunakan penghubung. Maka terjadi suatu integrasi sistem yang membentuk suatu kesatuan.

5. Masukan Sistem (*input*)

Input merupakan energi yang dimasukkan ke dalam suatu sistem dan dapat disebut juga dengan masukan sistem. Dapat diberi contoh di sistem komputer terdapat program yang merupakan *maintenance input* yang digunakan untuk operasi komputer. Sedangkan data merupakan *signal input* yang akan diolah menjadi suatu informasi.

6. Keluaran Sistem (*output*)

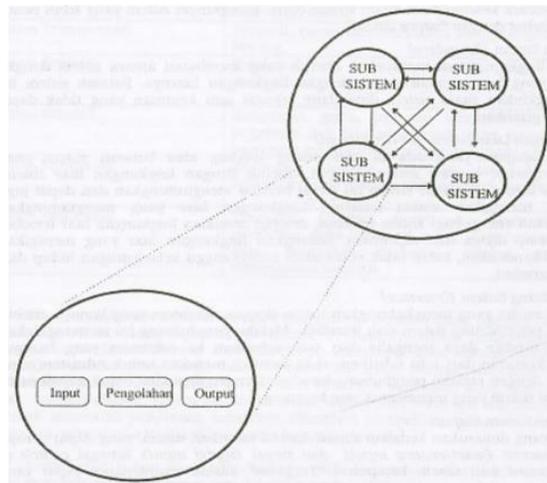
Merupakan suatu hasil dari energi yang diolah dan dapat diklasifikasikan menjadi *output* yang berguna. Keluaran sistem atau *output* dapat menjadi masukan atau *input* untuk subsistem lain. Keluaran sistem dapat digunakan untuk pengambilan keputusan.

7. Pengolah Sistem (*process*)

Suatu sistem memiliki proses yang dapat merubah masukan menjadi keluaran.

8. Sasaran Sistem

Sistem harus memiliki tujuan dan sasaran yang pasti, sistem yang berhasil apabila sistem dapat mengenai tujuan yang telah direncanakan.



Gambar 2.1 Karakteristik Sistem

(Sumber: Analisis Sistem Informasi [6, p.14])

2.2.2 Klasifikasi Sistem

Sistem merupakan bentuk integrasi satu unsur atau komponen dengan yang lain. Setiap sistem memiliki target sasaran yang berbeda tiap kasus yang terjadi pada sistem tersebut. Maka sistem dapat diklasifikasikan sebagai berikut : [6, p.15]

1. Sistem abstrak dan fisik

Sistem abstrak merupakan sistem yang berupa pemikiran dan tidak tampak secara fisik, sedangkan dengan sistem fisik merupakan sistem yang dapat terlihat secara fisik.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam dan tidak dibuat oleh manusia. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan manusia dengan mesin, dapat diberi contoh seperti sistem informasi yang melibatkan interaksi manusia dengan komputer.

3. Sistem deterministik dan sistem probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang dapat diprediksi, contoh program komputer yang dapat diprediksi tingkah lakunya. Sedangkan sistem probabilistik merupakan sistem yang kondisi masa depannya tidak dapat dilakukan prediksi dikarenakan mengandung probabilitas.

4. Sistem terbuka dan sistem tertutup

Sistem terbuka merupakan sistem yang berhubungan dan dapat dipengaruhi oleh lingkungan luar yang menerima *input* dan mengeluarkan *output* untuk subsistem lainnya. Sedangkan sistem tertutup merupakan sistem yang tidak dapat dipengaruhi oleh lingkungan luar. Sistem ini berjalan otomatis.

2.3. Informasi

Informasi adalah hasil dari pengolahan data yang menjadi bentuk yang lebih berguna dan bermanfaat bagi penerimanya. Data merupakan sumber informasi, data merupakan kenyataan atau fakta yang menggambarkan suatu peristiwa dan kesatuan nya nyata. [7, p.9]

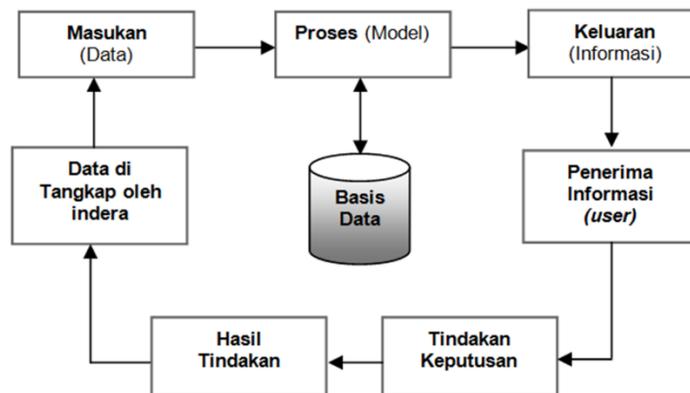
Menurut Gordon B. Davis (1995), informasi merupakan data yang telah diolah menjadi bentuk yang penting bagi penerima juga memiliki nilai yang nyata dalam keputusan-keputusan sekarang atau keputusan-keputusan yang akan datang. [7, p.9]

Informasi memiliki fungsi utamanya yaitu untuk menambah pengetahuan atau untuk mengurangi ketidakpastian penerima informasi. Informasi berguna untuk memberikan gambaran mengenai suatu permasalahan yang berdampak pada pengambilan keputusan yang dapat lebih cepat juga memberikan standar dan aturan bagi pihak pengambil keputusan. [7, p.9]

2.3.1. Siklus Informasi

Dengan menggunakan model proses tertentu, data diolah menghasilkan informasi. Dapat diberi contoh seperti suhu dalam fahrenheit diubah menjadi celcius. Dalam hal ini menggunakan model matematik berupa rumus konversi dari derajat fahrenheit menjadi celcius. Data diolah melalui suatu model menghasilkan informasi, kemudian penerima menerima informasi yang menghasilkan keputusan dan melakukan tindakan lain yang membuat data kembali. Data akan kembali ditangkap sebagai *input* dan diproses

kembali melalui suatu model proses dan seterusnya maka ini disebut sebagai siklus informasi. [7, p.10-11]



Gambar 2.2 Siklus Informasi

(Sumber : Konsep Sistem Informasi [7, p.11])

2.4. Sistem Informasi

Sistem Informasi adalah sekumpulan dari komponen-komponen pembentuk sistem yang memiliki hubungan antar satu komponen dengan komponen yang lain guna untuk menghasilkan informasi dalam bidang tertentu. [9, p.200]

Sistem Informasi juga dapat dikatakan sebagai sekumpulan *hardware*, *software*, *brainware*, prosedur, dan aturan yang diorganisasikan untuk mengolah data menjadi informasi yang bermanfaat bertujuan untuk memecahkan masalah dan mengambil keputusan. Sistem Informasi terbagi menjadi beberapa jenis, yaitu : [8, p.200-201]

1. *Transaction Processing System* (TPS)
2. *Office Automation System* (OAS)
3. Sistem Informasi Manajemen (SIM)

4. *Decision Support Systems* (DSS)
5. Sistem Ahli (ES)
6. *Group Decision Support Systems* (GDSS) dan *Computer-Support Collaborative Work Systems* (CSCW)
7. *Executive Support Systems*

2.4.1. Komponen Sistem Informasi

Terdapat beberapa komponen pembentuk sistem informasi yang disebut dengan blok bangunan. Semua komponen saling berhubungan dan berinteraksi satu komponen dengan komponen yang lain dan membentuk suatu kesatuan untuk mencapai sasaran. Berikut merupakan komponen-komponen sistem informasi : [9, p.202-203]

1. Komponen *Input*

Komponen ini mewakili data yang masuk dalam sistem informasi. *Input* berupa metode dan media untuk menangkap data yang dimasukkan.

2. Komponen Model

Komponen ini dapat berupa kombinasi prosedur, logika, dan model matematik yang melakukan manipulasi data masukan / *input* dan data yang disimpan di *database* (basis data) dengan cara yang telah ditentukan untuk dapat menghasilkan *output* yang diinginkan.

3. Komponen *Output*

Komponen ini berupa hasil keluaran dari sistem informasi yaitu berupa informasi yang berkualitas dan bermanfaat bagi pengguna sistem.

4. Komponen Teknologi

Komponen ini digunakan untuk menerima *input*, mengeksekusi model, menyimpan data, dan menghasilkan keluaran serta membantu pengendalian dari sistem.

5. Komponen *Hardware*

Komponen ini berperan penting karena memiliki peran sebagai media penyimpanan bagi sistem informasi. Memiliki fungsi sebagai penampungan sumber data dan informasi untuk membantu jalannya sistem informasi.

6. Komponen *Software*

Komponen ini memiliki peran sebagai tempat untuk mengolah, menghitung, dan melakukan manipulasi data yang didapat dari *hardware* untuk menghasilkan informasi,

7. Komponen Basis data

Komponen basis data ini berupa kumpulan data yang saling berhubungan dan berelasi satu dengan yang lain. Disimpan di *hardware* dan untuk menggunakan dapat dengan perangkat lunak. Data disimpan dalam basis data untuk keperluan penyediaan informasi.

8. Komponen Kontrol

Pengendalian perlu diterapkan guna meyakinkan bahwa hal-hal dapat merusak keberlangsungan sistem dapat dicegah. Apabila ditemukan kesalahan dapat ditangani lebih cepat.

2.5. Rekrutmen

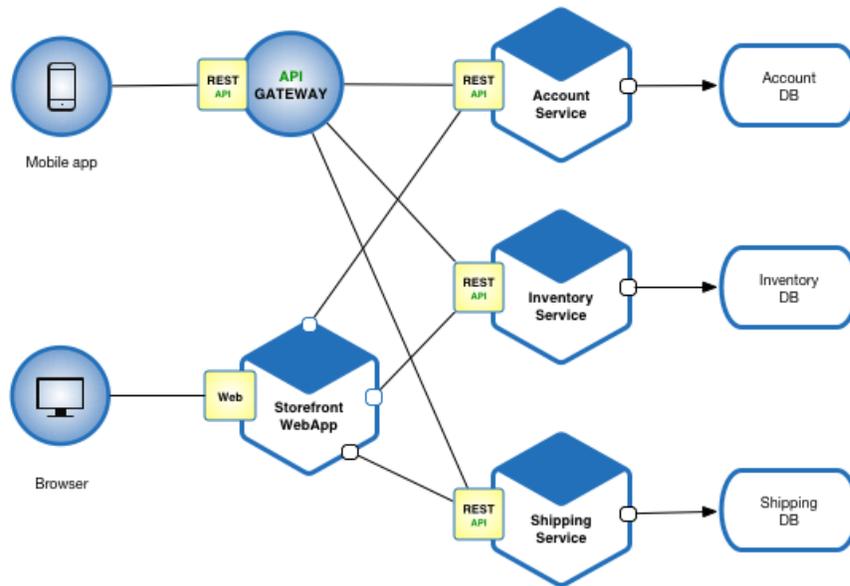
Menurut Simamora (2014), rekrutmen adalah rangkaian aktivitas mencari dan memikat pelamar kerja dengan motivasi, kemampuan yang dimiliki, keahlian yang dimiliki, dan pengetahuan yang dimiliki guna menutupi kekurangan yang diidentifikasi dalam perencanaan kepegawaian. Sedangkan menurut Rivai (2009), rekrutmen merupakan rangkaian kegiatan yang dimulai dengan kebutuhan perusahaan akan tenaga kerja dan membuka lowongan sampai berhasil mendapatkan calon yang memenuhi kualifikasi sesuai dengan lowongan yang ada. [3, p.16]

Rekrutmen menjadi masalah yang penting bagi perusahaan dalam hal pengadaan tenaga kerja. Rekrutmen dinilai berhasil dengan kata lain banyak pelamar yang tertarik dan melamar, maka peluang untuk perusahaan akan mendapatkan karyawan yang terbaik akan semakin lebar. Perusahaan dapat leluasa untuk melakukan seleksi dari pilihan yang ada. Setiap pelaksanaan rekrutmen akan dilakukan pelatihan, pelatihan dilakukan guna menambah pengetahuan keterampilan untuk karyawan dapat melakukan pekerjaan tertentu. [3, p.17]

2.6. Arsitektur *Microservice*

Arsitektur *Microservice* adalah desain arsitektur perangkat lunak yang memecah aplikasi atau sistem menjadi *service-service* kecil yang terpisah sesuai dengan fungsinya. Fungsi dari pembagian menjadi *service-service* kecil ini memberi dampak sistem dapat bekerja secara independen. Setiap *service* dapat mempunyai teknologi atau bahasa pemrograman yang berbeda antara satu *service* dengan *service* lain yang

berarti terdapat teknologi berbeda dalam satu sistem. Dalam arsitektur *microservice* ini antar *service* saling berkomunikasi satu *service* dengan yang lain. [10]



Gambar 2.3 Arsitektur *Microservice*

(Sumber : www.microservices.io)

Arsitektur *microservice* dapat dilakukan pemecahan antar service secara bisnis.

Terdapat beberapa kelebihan dari arsitektur *microservice*, yaitu : [11]

1. Memudahkan dalam melakukan *scalability*.
2. Pengembang aplikasi atau sistem memiliki kebebasan dalam mengembangkan aplikasi atau menambahkan modul baru dengan bahasa pemrograman dan *framework* apapun.
3. Memudahkan pemeliharaan perangkat lunak.
4. Dapat melakukan proses distribusi perangkat lunak secara independen.

5. Kode aplikasi dapat lebih sedikit dan bersifat independen sehingga dapat dilakukan pengujian secara independen.

2.6.1 Domain Design Driven

Domain Design Driven adalah suatu pendekatan perangkat lunak dalam model atau pemecahan aplikasi. *Domain Design Driven* membantu dalam proses pengembangan perangkat lunak menjadi akurat sesuai dengan proses bisnis yang akan berjalan di aplikasi. Salah satu masalah sering terjadi ketika pembangunan aplikasi yaitu dengan seiringnya berkembangnya aplikasi atau sistem yang dibangun maka akan semakin besar. Masalah ini memberi menimbulkan *dead code* yakni ketika kode aplikasi sudah besar maka pengembang merasa takut untuk menambah fitur ataupun modul karena takut menimbulkan kesalahan [10].

2.7. Application Programming Interface (API)

Application Programming Interface adalah sebuah *interface* (antarmuka) yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. *Application Programming Interface* (API) juga dapat dikatakan sebagai sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh pengembang saat membangun perangkat lunak. Keunggulan dari *application programming interface* (API) ini yaitu memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. [12]

2.7.1. REST API

REST API (*Representational State Transfer*) merupakan suatu prinsip arsitektur untuk melakukan transmisi data melalui antarmuka yang terstandarisasi seperti HTTP. REST API ini bekerja seperti web biasa. *Client* mengirimkan *HTTP request* dan kemudian *server* memberikan *response* kepada *client*. REST dikembangkan oleh Roy Fielding yang merupakan co-founder dari Apache HTTP Server Project. Arsitektur REST ini terdapat enam batasan, yaitu : [12]

1. *Client Server*

Batasan *client server* merupakan penjelasan suatu antarmuka yang memisahkan bagian *client* dan *server*. Melalui pemisahan antarmuka, REST memberi keuntungan untuk *client* yang tidak perlu berurusan dengan masalah penyimpanan. Hal ini dapat meningkatkan portabilitas antarmuka pengguna diberbagai *platform* dan dapat meningkatkan *scalability*.

2. *Stateless*

Batasan ini menjelaskan bahwa *server* tidak menyimpan *state* atau penanda dari *client*. Setiap pesan yang dikirimkan oleh *client* bersifat *self-descriptive* yang berarti pesan yang dikirimkan oleh *client* memiliki informasi yang cukup untuk *server* dapat memproses pesan atau *request* tersebut. Setiap *state* atau *session* tersimpan di pihak *client*.

3. *Cacheable*

Batasan ini menjelaskan bahwa *response* dari *server* memiliki sifat *cacheable*. Setiap *response* server dapat disimpan secara implisit, eksplisit, atau *negotiated*.

4. *Uniform Interface*

Uniform Interface menjelaskan antarmuka antara *client* dan *server*. Hal ini memisahkan setiap arsitektur kedua pihak dapat berkembang secara independen. Batasan ini merupakan hal fundamental untuk RESTful. REST API ini menggunakan HTTP *method* (GET, POST, PUT, DELETE) untuk menjelaskan metode permintaan. *Uniform Resource Identifier* (URI) untuk mengidentifikasi nama sumber, HTTP *response* untuk menjelaskan informasi yang dikembalikan oleh *server*.

5. *Layered System*

Batasan ini menjelaskan bahwa pihak *client* tidak bisa secara langsung terhubung dengan *server*. Terdapat perantara antar *client* dengan *server*. Perantara tersebut untuk meningkatkan *scalability* dengan memungkinkan *load-balancing* dan menyediakan *cache*. *Layered system* dapat menerapkan kebijakan keamanan.

6. *Code On Demand*

Batasan ini menjelaskan *server* dapat memberikan atau menyesuaikan fungsionalitas *client* sementara dengan mentransferkan *logic* kedalamnya sehingga dapat dijalankan.

REST merupakan gaya arsitektur untuk penerapan *web service* dalam menerapkan konsep perpindahan antar *state*. Kelebihan menggunakan REST yaitu interaksi REST menggunakan *Hyper Text Transfer Protocol (HTTP)* yang merupakan hal yang sudah umum digunakan. [12]

2.8. Express.js

Express js merupakan sebuah *framework* web dari Node.js. Node js tersendiri merupakan *runtime environment* untuk bahasa pemrograman javascript agar dapat dijalankan di *server-side*. *Framework* ini dapat digunakan untuk mengembangkan produk aplikasi web maupun REST API. [13]

2.9. Laravel

Laravel merupakan *framework* web berbasis PHP dan menerapkan paradigma *object oriented programming (OOP)*. Laravel ini merupakan *framework open source* yang diciptakan oleh Taylor Otwell dan dibangun untuk pengembangan web yang menggunakan pola MVC. [14]

2.10. MySQL

MySQL adalah *Database Management System (DBMS)* yang *open source* dengan dua bentuk lisensi. MySQL ini termasuk dalam jenis RDBMS (*Relational Database Management System*). Maka terdapat istilah pada RDBMS umumnya seperti baris, kolom, tabel, relasi. MySQL merupakan *database engine* yang mendukung bahasa SQL untuk melakukan pengelolaan datanya. MySQL juga merupakan DBMS yang *multithread, multi-user*. [15]

2.11. React.js

React.js merupakan *user interface library* yang dikembangkan oleh facebook (Meta) guna untuk memfasilitasi pengembang web agar dapat interaktif, *stateful*, dan komponen UI dapat *reusable*. React sudah digunakan pada *website* facebook dibagian *production*. React.js merupakan *library* terbaik untuk melakukan *render* UI yang kompleks dengan memberikan performa yang tinggi. Konsep yang digunakan pada react.js ini yaitu *virtual dom*. [16]