# BAB II

**LANDASAN TEORI**

## Definisi Sistem, Informasi, dan Sistem Informasi

### Sistem

Menurut James A. O’Brien dan George M.Marakas, ( 2011 : 26 )Sistem adalah satu paket komponen yang saling terhubung, dengan parameter yang jelas, bekerja sama untuk mecapai seperangkat tujuan bersama dengan menerima masukan dan menghasilkan output dalam sebuah proses transformasi yang terorganisir.

### Informasi

Menurut James A. O’Brien dan George M.Marakas, ( 2011 : 12).Informasi adalah data yang akan berguna bagi pengguna tertentu dan data tersebut sudah dikonversikan menjadi konteks yang memiliki arti .

### Sistem Informasi

Menurut James A. O’Brien dan George M.Marakas, (2011 : 4 ).Sistem informasi (SI) dapat berupa gabungan antara seseorang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, kebijakan, dan prosedur yang menyimpan, mngambil, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Komponen sistem informasi terbagi atas beberapa hal yaitu sumber daya manusia (sebagai pengguna dan ahli SI) , sumber daya perangkat keras (mesin atau media), smber daya perangkat lunak (sebagai program dan prosedur), sumber daya data (sebagai data dan pengetahuan), sumber daya jaringan (sebagai media komunikasi dan dukungan jaringan).

## *Unified Modeling Language* (UML)

Menurut Martina Seidl et al, (2014 : 15).*Unified Modeling Language* (UML) adalah konsolidasi praktik terbaik yang telah ada selama bertahun-tahun dalam penggunaan bahasa pemodelan. UML mempermudah kita dalam menyajikan berbagai aspek sistem perangkat lunak yang bervariasi (misalnya , persyaratan,struktur data,arus ata, dan arus informasi) dalam kerangka tunggal yang menggunakan konsep berorientasi objek.

Dalam UML, sebuah model akan di representasikan secara terstruktur dalam bentuk diagram. Sebuah diagram menjelaskan tentang bagian realita yang digambarkan oleh model. Ada diagram yang dapat memberikan gambaran tentang pengguna mana yang menggunakan fungsionalitas dan diagram yang menjelaskan struktur sistem tanpa implementasi yang konkret. Ada juga diagram yang mewakili proses yang diperbolehkan dan dilarang.

### *Use Case Diagram*

Menurut Martina Seidl et al (2014 : 15)UML menawarkan *diagram* *use case* untuk menentukan persyaratan yang harus dipenuhi oleh sistem. Diagram ini menjelaskan pengguna mana yang menggunakan fungsi sistem namun tidak menjelaskan rincian spesifik penerapannya. Unit fungsionalitas yang disediakan sistem untuk penggunanya disebut use case.

### *Activity Diagram*

Menurut Martina Seidl et al, (2014 : 20).*Activity diagram* berfungsi untuk memodelkan proses apa pun menggunakan diagram aktivitas: proses bisnis dan proses perangkat lunak. *Activity diagram* menawarkan mekanisme kontrol aliran serta mekanisme arus data yang mengkoordinasikan tindakan yang membentuk suatu aktivitas.

### *Sequence Diagram*

Menurut Martina Seidl et al, (2014 : 20). *Sequence diagram* menggambarkan interaksi antara objek untuk memenuhi tugas tertentu. Fokusnya adalah pada urutan kronologis pesan yang dipertukarkan di antara pasangan interaksi. Berbagai konstruksi untuk mengendalikan urutan kronologis pesan serta konsep modularisasi yang memungkinkan untuk memodelkan interaksi kompleks.

### *Class Diagram*

Menurut Martina Seidl et al, (2014 : 17). Konsep *class diagram* berasal dari pemodelan data konseptual dan pengembangan perangkat lunak berorientasi objek. Konsep ini digunakan untuk menentukan struktur data dan struktur objek suatu sistem. *Class diagram* didasarkan terutama pada konsep kelas, generalisasi, pdan asosiasi.

## *Prototyping Model*

Menurut Car dan Verner dalam buku Pedro Isaias dan Tomayess Issa, 2015 : 33).Model *prototyping* adalah kerangka kerja berulang yang menjadi pusat dari banyak pendekatan yang lebih cepat untuk pengembangan perangkat lunak, sejak awal 1980-an. Pada 1997, Carr dan Verner mengamati bahwa model prototyping lebih dinamis dan lebih responsif terhadap kebutuhan klien, serta lebih efisien. seperti juga kurang berisiko dan lebih efisien. Karena alasan itu, mereka berusaha merangkum model *prototyping* dalam satu kerangka kerja yang konsisten.

Model *prototyping* digunakan untuk menciptakan sistem yang masih versi percontohan (sample), yang disebut prototype. Tujuannya adalah untuk membangun berbagai versi dan secara konsisten memperbaiki versi-versi tersebut hingga produk akhir tercapai, Carr dan Verner (1997). Pembuatan versi – versi sistem tersebur lebih ditekankan kepada respon atau umpan balik pengguna, hal ini sangat mendasar untuk mengembangkan *prototype* berikutnya, hingga produk akhir.

Produk prototipe, yang telah dibangun sehingga dapat diuji secara langsung dan pengguna dapat memberikan umpan balik dan pengalaman real-time. Jika pengguna memberikan masukan yang di anggap perlu untuk, *prototype* akan direvisi dan disempurnakan menjadi versi yang lebih baru. Sub siklus ini akan berlangsung hingga produk diterima secara umum oleh pengguna dan tidak lagi membutuhkan perubahan besar atau pembaruan, pada saat mana versi final dirilis.

Model *prototyping* juga memiliki kelebihan dan kelemahan sebagai berikut:

### Kelebihan Model *Prototyping* adalah sebagai berikut:

#### Meningkatnya keterlibatan pengguna dalam produk bahkan sebelum implementasinya.

#### Karena model kerja sistem ditampilkan, pengguna mendapatkan pemahaman yang lebih baik tentang sistem yang sedang dikembangkan.

#### Mengurangi waktu dan biaya karena kegagalan dapat dideteksi jauh lebih awal.

#### Review pengguna yang lebih cepat mengarah ke solusi yang lebih baik.

#### Fungsi yang hilang dapat dikenali dengan mudah.

#### Fungsi membingungkan atau sulit dapat diidentifikasi.

### Kelemahan model prototyping adalah sebagai berikut:

#### Risiko analisis kebutuhan yang tidak mencukupi karena terlalu banyak ketergantungan pada prototype.

#### Pengguna mungkin bingung dalam prototype dan sistem sebenarnya.

#### Praktis, metodologi ini dapat meningkatkan kompleksitas sistem karena ruang lingkup sistem dapat berkembang melampaui rencana awal.

#### Pengembang dapat mencoba untuk menggunakan kembali prototype yang ada untuk membangun sistem yang sebenarnya, walaupun tidak layak secara teknis.

#### Upaya yang diinvestasikan dalam membangun prototype ungkin terlalu banyak jika tidak dipantau dengan benar.

## *8 Golden Rules User Interface*

## Pada perancangan antarmuka web *Optical Character Recognition* ini menerapkan kaidah perancangan ”8 Golden Rules User Interface Design” yang telah disusun, Menurut Ben Shneidrman ( 2016 ). Berikut adalah penjabarannya :

1. Strive for consistency

Konsistensi dibutuhkan antar halaman dalam satu aplikasi ataupun antara aplikasi yang masih berhubungan. Gunanya adalah supaya user, terutama novice user, tetap dapat mengenali halaman yang dilihat masih dalam lingkup atau masih memiliki hubungan dengan aplikasi yang digunakan.

1. Cater to universal usability

Dalam merancang antarmuka aplikasi, seorang interface designer harus memperhitungkan jenis variasi user nya. Baik itu dari segi latar belakang budaya dan bahasa, juga variasi tingkat pemahaman user terhadap aplikasi. Pada poin ini yang lebih sering dipikirkan adalah perbedaan kebudayaan user, sehingga aplikasi harus dirancang dalam berbagai macam bahasa.

1. Offer informative feedback

Informative feedback tidak harus selalu dengan jawaban dari aplikasi ke user, tetapi dapat berupa perubahan antarmuka setiap user melakukan aksi, dengan demikian user paham bahwa aksinya sudah direspon oleh aplikasi.

1. Design dialogs to yield closure

Untuk poin ini sebenarnya termasuk dari bagian informative feedback, dengan menyampaikan bahwa proses yang dijalankan oleh user sudah selesai.

1. Prevent errors

Poin ini untuk menjaga agar user tidak melakukan kesalahan dalam menjalankan proses. Sangat diperlukan supaya user tidak merasa jenuh dalam mencoba menggunakan aplikasi, karena tidak bisa menemukan format / aksi yang tepat pada saat mencoba suatu fungsi.

1. Permit easy reversal of actions

Poin ini merupakan salah satu poin yang cukup penting untuk menunjang UX dari suatu aplikasi. Biasanya yang dianggap sebagai pemenuhan poin ini adalah tombol back.

1. Support internal locus of control

Poin ini terutama sangat disukai oleh user yang sudah terbiasa menggunakan aplikasi, karena biasanya user ingin memiliki tampilan yang bisa diatur oleh user sendiri sesuai preferensi dari user.

1. Reduce short-term memory load

Pada poin ini biasanya orang lebih memusatkan pada desain tata letak menu dan tombol. Tetapi sebenarnya akan lebih efektif jika diterapkan pada proses saat user harus memberikan input ke sistem.

## *Optical Character Recognition* (OCR)

### Menurut Suryo Hartanto, (2015 : 16). *Optical Character Recognition* (OCR) adalah sebuah aplikasi komputer yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk dikonversi ke dalam bentuk file tulisan. Sistem pengenal huruf ini dapat meningkatkan fleksibilitas atau kemampuan dan kecerdasan sistem komputer.

Secara umum proses OCR memiliki beberapa tahap, berikut adalah tahap – tahapnya :

#### *File Input*

Menerima inputan dari user yang merupakan foto yang akan diproses dalam format ( PDF, screenshoot, JPG)

#### *Prepoccessing*

Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya.

#### Segmentasi

Segmentasi adalah proses memisahkan area pengamatan (region) pada tiap karakter yang dideteksi.

#### Normalisasi

Normalisasi adalah proses merubah dimensi region tiap karakter dan ketebalan karakter.

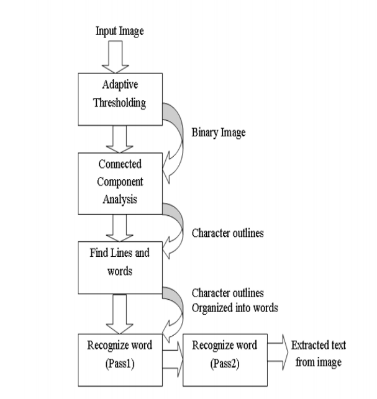
#### Ekstrasi

### Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.

#### *Recognition*

*Recognition* merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.

### *Optical Character Recognition Arsitekture*



Gambar 2‑1: Optical Character Recognition Architecture

Sumber : Onyejegbu L. N. Department of Computer Science (N., 2016)

*Input Image* : Sistem OCR menerima input berupa gambar

*Adaptive* *Thresholding* : meng-skalasi ulang warna dari gambar sebagai sebuah input.

*Connected* *component* *Analysis* : Menghubungkan masing – masing komponen yang telah di analisis dari *Adaptive* *Thresholding*.

*Find* *Lines* and *Words* : Menentukan masing – masing baris dan menentukan dimana huruf – huruf tersebut harus diletakan.

*Recognize* *Word* : mengidentifikasi huruf dan kata yang telah diterjemahkan dari *binary* menjadi huruf yang dapat dikenali manusia.

### Teknik – teknik dalam OCR

Dengan muncul teknologi ini tentunya terdapat teknik – teknik yang digunakan dalam memproses foto – foto yang di-input. Berikut merupakan bberapa teknik dalam Optical Character Recognition :

#### *Matriks* *Macthing* : Matriks matching mengubah setiap karakter kedalam pola maktriks, dan kemudian membandingkan pola dengan index karakter yang dikenal.

#### *Fuzzy* *Logic* : *fuzzy* *logic* adalah logika multi-nilai yang dapat membantu komputer dalam mendefinisikan antara beberapa evaluasi konvensional seperti, ya atau tidak, hitam atau putih. *Fuzzy* *Logic* membangun pola pikir yang mengarah kepada manusia.

#### *Neural* *Network* : strategi ini menirukan cara kerja sistem syaraf manusia, masing – masing layer akan mencocokan bentuk – bentuk yang terdapat pada gambar dan mencocokannya dengan bentuk yang mereka kenal dengan ‘pola pixel karakter”, OCR dengan Neural Network sangat membantu dalam membaca dokumen atau gambar yang rusak.

## *Artificial* *Intelligence*

Menurut Jan Romportl Eva, (2015 : 23).Kecerdasaan buatan adalah mesin yang bekerja sesuai hipotesis, kecerdasaan buata sebelumnya belum pernah ada di planet ini dan berisi tentang karakteristik yang kita miliki tetapi dengn penggambaran yang masih belum jelas.

Menurut Hendra, (2015 : 12).*Artificial* *Intelligence* adalah salah satu bagian ilmu komputer yang membuat agar mesin (komputer)melakukan pekerjaan seperti manusia, serta berkembang sebaik manusia.

Selain itu, Hendra juga membandingkan Kecerdasan Buatan dengan Kecerdasan alami dan dari beberapa keuntungan Artificial Intelegence dapat di artikan bahwa proses Optical Character Recognition merupakan salah satu Artificial Inteligence. Kecerdasan Buatan memiliki beberapa keuntungan sebagai berikut:

### Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dikarenakan manusia masi memiliki memori yang dapat lupa akan sesuatu hal ini didasarkan oleh sifat manusia itu sendiri.Kecerdasan buatan tidak akan berubah selama sistem komputer & komputer tidak mengubahnya.

### Kecerdasan buatan lebih mudah diduplikasi dan disebarkan. Mentransfer pengetahuan manusia dari seseorang ke orang lain tidak akan semaksimal kecerdasan buatan, serta membutuhkan proses yang cukup lama.

### Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan harus mendatangkan seseeorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

### Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

### Kecerdasan buatan dapat didokumentasi dan direproduksi lebih mudah. Kecerdasan alami sangat sulit untk direproduksi dan di dokumentasikan.

### Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.

## *React*

Menurut Joel Vainikka, (2015 : 17). *React* (juga disebut sebagai *ReactJS*) adalah library *JavaScript* open-*source* untuk membangun situs web dinamis antarmuka pengguna. Komponen React ditulis dengan *JavaScript* XML (JSX) pra-prosesor untuk memformat antarmuka pengguna. *Sintaks* *React* adalah kombinasi kode JavaScript dan HTML. Kode ini kemudian dikompilasi ke JavaSc*r*ipt.

Keputusan platform untuk aplikasi dari sisi klien dipersempit menjadi dua pilihan, yaitu Angular dan *React*. *React* mempunyai opsi yang lebih baik dan menarik untuk dikembangkan serta untuk mempelajari sesuatu yang baru.

## Node.JS

Menurut Muhammad Rizki Samsul Ariefin, (2014 : 22). Node Js adalah sebuah platform untuk membangun aplikasi secara real-*time*.Node Js dapat menangani aksi *event* seperti *production*, *detection* , *reaction*, dan *consumption* pada server secara non-blocking yang berarti data - data yang diterima server bisa dimanipulasi atau terdistribusi keclient tanpa disimpan terlebih dahulu.

## *Javascript* *Object* *Notation* (JSON)

Menurut Samuel Aji Sena, (2013 : 2). JSON (*JavaScript* *Object* *Notation*) merupakan sarana untuk memasukan data kedalam sebuah variable. Sangat mudah digunakan dan dipahami oleh manusia, dan mudah juga untuk komputer dalam melakukan parsingnya. JSON merupakan bagian dari bahasa pemrograman *JavaScript* (Standard ECMA-262 3rd *Edition* –*December* 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C. Inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal.JSON dibangun dalam dua struktur :

### Beberapa pasangan dari nama dan nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah *object*, *record*, *struct*, *table* *hash*, *key* *list* atau *associative* *array*.

### Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan array, *vector*, *list* atau daftar dalam bahasa pemrograman.

JSON dikatakan sebagai bahasa universal karena bisa digunakan hampir pada semua bahasa pemrograman. (Dwija Wisnu Brata, 2015)

JSON adalah sebuah media yang digunakan sebagai pertukaran data pada internet. Berkas JSON berbasis teks dan mudah dipahami, hal ini membuat json mudah dikenali oleh berbagai macam bahasa pemrograman ssehingga sangat ideal untuk digunakan sebagai media pertukaran data antar aplikasi yang berbeda bahasa pemrogramannya.

## *Application* *Programming* *Interface* (API)

Menurut M. Syaiful Amri, (2015 : 7). API atau *Application* *Programming* *Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan programmer untuk “membongkar” suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinka programmer menggunakan sistem function. Proses ini dikelola melalui operating system. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi.

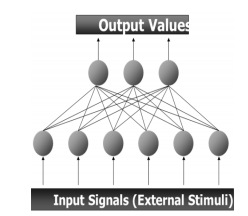
Menurut Samuel Aji Sena (2013 : 10), *Appication* *Programming* *Interface* (API) adalah merupakan perintah dan protocol yang digunakan untuk menghubungkann sebuah aplikasi dengan aplikasi lain agar dapat saling berinteraksi. Seiring dengan perkembangan internet, API dapat diimplementasikan pada sisi server dan dapat digunakan oleh beberapa aplikasi yang dapat terhubung ke server dengan menggunakan protokol tertentu. Padaprotokol *HTTP, Appication Programming Interface* umumnyadisebutsebagai *Web Appication Programming Interface Server* atau *Web Service.*

## Penelitian Terdahulu

1. *Optical* *Character* *Recognition* *Tecnique* *Algorithm* ( *Neural* *Network*)

*Neuralnetwork* adalah *tools* pemodelan data yang kuat dan mampu menghasilkan dan merepresentasikan *input/output* yang kompleks. Ide terbentuknya *Neural* *Network* awalnya bertujuan untuk membuat sistem dapat melakukan perhitungan cerdas yang serupa untuk dilakukan oleh otak manusia.

berikut dalah gambaran dasar Model *Neural* *Network*:



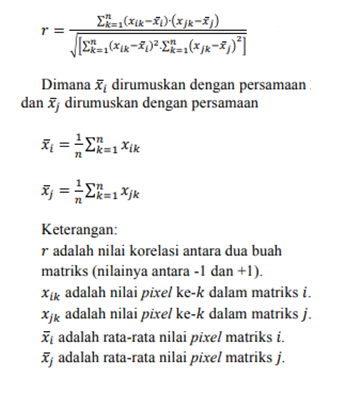
Gambar 2‑2 : *Multiplayer Neural Network*

Model *Neural* *Network* yang paling umum adalah persepsi *multiplayer*, jenis jaringan ini merupakan jaringan yang dapat berkembang dengan sendirinya. Model yang memetakan input kepada *output* yang benar menggunakan data hitoris.

1. *Optical* *Character* *Recognition* Menggunakan Algoritma *Template* *Colleraltion*

*Template* *matching* adalah salah satu teknik dalam pengolahan citra digitalyang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan) Citra masukan dibandingkan dengan citra *template* yang ada di dalam basis data, kemudian dicari kesamaannya dengan menggunakan suatu aturan tertentu. Pencocokkan citra yangmenghasilkan tingkat kemiripan dan kesamaan yang tinggi.

Kesamaan antar dua buah matriks citra dapat dihitung nilainya dengan menghitung nilai korelasinya (*correlation*). Nilai korelasi dua buah matriks dapat dihitung dengan menggunakan rumus dibawah ini:



Gambar 2‑3 Rumus Matriks Macthing

Sumber : Optical Character Recognition For Indonesian Id-Card (2014 : 7)