

BAB II

TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan sebenarnya sudah dimulai sejak musim panas tahun 1956. Pada waktu itu sekelompok pakar komputer, pakar dan peneliti dari disiplin ilmu lain dari berbagai akademi, industri serta berbagai kalangan berkumpul di Dartmouth College untuk membahas potensi komputer dalam rangka menirukan atau mensimulasi kepandaian manusia. Pada mulanya kecerdasan buatan hanya ada di universitas dan laboratorium penelitian, menjelang akhir tahun 1970-an dan awal tahun 1980-an, mulai dikembangkan secara penuh dan hasilnya secara berangsur-angsur mulai dipasarkan (Victor Amrizal & Qurrotul Aini, 2013).

Definisi kecerdasan buatan atau *artificial intelligence* (AI), menurut beberapa pakar:

- a. Schalkoff (1990): AI adalah bidang studi yang berusaha menerangkan dan meniru perilaku cerdas dalam bentuk proses komputasi.
- b. Rich dan Knight (1991): AI adalah studi tentang cara membuat komputer melakukan sesuatu yang, sampai saat ini, orang dapat melakukannya lebih baik.
- c. Luger dan Stubblefield (1993): AI adalah cabang ilmu komputer yang berhubungan dengan otomasi perilaku yang cerdas
- d. Haag dan Keen (1996): AI adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi informasi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia.

Kecerdasan buatan memiliki beberapa bidang, diantaranya adalah sebagai berikut (Victor Amrizal & Qurrotul Aini, 2013):

- a. Sistem pakar (*expert system*): Sistem pakar adalah sebuah program komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar (*human expert*).
- b. *Computer Vision* : *Computer Vision* adalah ilmu dan teknologi mesin yang memiliki fitur untuk melihat, di mana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu.
- c. Logika Samar (*fuzzy logic*): *Fuzzy logic* menyediakan cara sederhana untuk menggambarkan kesimpulan pasti dari informasi yang ambigu, samar-samar atau tidak tepat.
- d. Jaringan saraf tiruan (*neural networks*): Jaringan saraf tiruan adalah salah satu bagian ilmu komputer yang membuat agar komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia
- e. Sistem Pendukung Keputusan: Secara umum DSS adalah sistem berbasis komputer yang interaktif, yang membantu mengambil keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang terstruktur.
- f. *Data Mining*: *data mining* merupakan gabungan sejumlah disiplin ilmu komputer yang didefinisikan sebagai proses penemuan pola-pola baru dari kumpulan-kumpulan data yang sangat besar, meliputi metode-metode yang merupakan irisan dari kecerdasan buatan (*artificial intelligence*).

2.2 Data Mining

Data mining atau penambangan data adalah teknik yang relatif cepat dan mudah untuk menemukan pengetahuan, pola dan / atau relasi antar data. Secara umum, kegunaan *data mining* dibagi menjadi dua yaitu deskriptif dan prediktif. Deskriptif berarti *data mining* digunakan untuk mencari pola-pola yang dapat dipahami manusia yang menjelaskan karakteristik

data. Sedangkan prediktif berarti *data mining* digunakan untuk membentuk sebuah model pengetahuan yang akan digunakan untuk melakukan prediksi (Dr. Suyanto 2019).

Berdasarkan fungsionalitasnya, tugas-tugas *data mining* bisa dikelompokkan ke dalam enam kelompok berikut ini (Dr. Suyanto 2019):

- a. Klasifikasi: men-generalisasi struktur yang diketahui untuk diaplikasikan pada data-data baru.
- b. Klasterisasi: mengelompokkan data, yang tidak diketahui label kelasnya, kedalam sebuah kelompok tertentu sesuai dengan ukuran kemiripannya.
- c. Regresi: menemukan suatu fungsi yang memodelkan data dengan galat (kesalahan prediksi) semaksimal mungkin.
- d. Deteksi anomali: mengidentifikasi data yang tidak umum, bisa berupa *outliner* (pencilan), perubahan atau deviasi yang mungkin sangat penting dan perlu investigasi lebih lanjut.
- e. Pemodelan kebergantungan: mencari relasi antar variabel
- f. Perangkuman: menyediakan representasi data yang lebih sederhana, meliputi visualisasi dan pembuatan laporan.

2.3 Metode *K-Nearest Neighbor*

Metode *Nearest Neighbor* dikenal juga dengan *lazy learner* (pembelajar malas) karena tidak ada proses belajar (dari data) untuk menghasilkan aturan. *Nearest Neighbor* hanya menyimpan data-data *training* dan kemudian belajar dari data-data tetangga (terdekat) secara langsung pada saat melakukan klasifikasi. Terdapat banyak metode yang termasuk ke dalam *Nearest Neighbor* salah satunya adalah *K-Nearest Neighbor* (Dr. Suyanto 2019).

Metode *K-Nearest Neighbor* (K-NN) bekerja dengan mencari sejumlah k objek data atau pola dari semua data *training* yang paling dekat dengan pola masukan, kemudian memilih kelas dengan jumlah pola terbanyak di antar k pola tersebut. Penentuan k pola terdekat dilakukan berdasarkan

pupuk hijau dan kompos yang mengandung unsur hara yang dibutuhkan tanaman untuk mencukupi kebutuhan hara yang diperlukan tanaman sehingga mampu berproduksi dengan baik. Pupuk merupakan kunci dari kesuburan tanah karena berisi satu atau lebih unsur untuk menggantikan unsur yang habis terisap tanaman (Lingga, 2002).

2.6 Basis Data (*Database*)

Basis Data (*Database*) adalah tempat penyimpanan data dan informasi secara terstruktur dan teratur, yang digunakan sebagai pangkalan data atau tempat berkumpulnya data secara digital. Pada *database* terdapat sebuah sistem yang disebut *DBMS (Database Management System)*, *DBMS* adalah sebuah sistem untuk membantu aplikasi dan pengguna di dalam melakukan manajemen data dan *database*. Beberapa contoh *DBMS* yang umum digunakan di dunia komputer, antara lain adalah *MySQL*, *PostgreSQL*, *Oracle*, *Microsoft SQL Server*, *Maria DB*, *Mongo DB*, dan lainnya, (I Putu Agus Eka Pratama, 2018).

2.7 Definisi *PostgreSQL*

PostgreSQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian lisensi BSD. Perangkat lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain *MySQL* dan *Oracle*. *PostgreSQL* adalah sistem *database* yang kuat untuk urusan relasi, *open source*. Memiliki lebih dari 15 tahun pengembangan aktif dan sudah terbukti segala rancangan arsitekturnya telah mendapat reputasi tentang kuat, handal, integritas data, dan akurasi data, (Regina Obe, Leo Hsu, 2016).

2.8 *Python*

Python adalah bahasa pemrograman *multi-platform* yang bersifat *free* dan *open-source*, dan dapat digunakan untuk mengembangkan aplikasi-

aplikasi *desktop* maupun *web*. Python memiliki pustaka standar (*Python Standard Library*) yang sangat lengkap sehingga dapat memenuhi berbagai macam permasalahan-permasalahan didalam dunia pemrograman, sebagai alternatif dari bahasa-bahasa pemrograman lain seperti C, C++, *Java*, *PHP*, dll. Untuk kepentingan yang spesifik, kode *python* juga dapat diintegrasikan dengan pustaka lain yang ditulis di dalam bahasa C, C++, *Java* (melalui *Jython*), dan bahasa-bahasa .NET seperti Visual Basic dan C# (melalui *IronPython*), (Budi Rahardjo, 2019).

2.9 *JavaScript*

JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan *website* agar lebih dinamis dan interaktif. Dengan *JavaScript programmer* dapat membuat *game*, animasi dan bentuk interaktif-interaktif lain didalam *web* (Priyanto Hidayatullah & Jauhari Khairul Kawistara, 2020).

2.10 *Jquery*

JQuery adalah kumpulan kode/fungsi *Javascript* siap pakai, sehingga memudahkan dan mempercepat dalam membuat kode *Javascript*. *JQuery* menyederhanakan kode *Javascript*, (Faizal Ari Prabowo, Mamay Syani, 2017).

2.11 *Odoo Framework*

Odoo Framework merupakan sebuah platform *open source* yang digunakan untuk keperluan bisnis. Aplikasi atau modul-modul yang terintegrasi dibangun di atas platform tersebut, meliputi semua bisnis dari CRM, Sales, stok, dan lain sebagainya. *Odoo* dibangun menggunakan bahasa pemrograman *Python*, *XML*, dan *JavaScript*. *Odoo* termasuk sebagai *software Enterprise Resource Planning (ERP)*. *Odoo* dulunya dikenal sebagai *OpenERP*. *Odoo* dibangun secara *open source*, sehingga *Odoo* mendukung pemanfaatan kembali *library* yang telah ada dan setiap

orang dapat terlibat dalam pengembangannya. Platform Odoo terdiri dari tiga komponen utama, yaitu *database PostgreSQL* sebagai *database* bawaannya, *application server* Odoo, dan *web server*. *Database PostgreSQL* menampung semua data yang berhubungan dengan data dan konfigurasi Odoo. Selain sebagai aplikasi, Odoo juga dapat berfungsi sebagai *Framework* atau kerangka kerja bagi para *Software Developer*, (Daniel Reis, 2019).

Odoo menyediakan modul-modul dasar yang mendukung fungsi bisnis, sehingga setiap modul dapat dikostumisasi sesuai dengan kebutuhan. Modul-modul tersebut terdiri dari 8 kelompok aplikasi, antara lain:

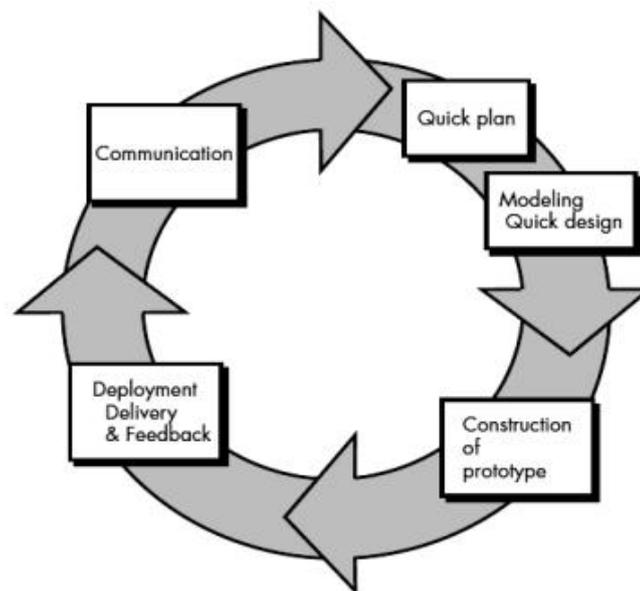
1. Aplikasi *website* : *website builder, blog, e-Commerce, forum, slides, live chat, appointments.*
2. Aplikasi *sales management* : *Customer Relationship Management (CRM), point of sale, Sales, subscriptions.*
3. Aplikasi keuangan : *accounting, invoicing, expenses.*
4. Aplikasi operasi : *inventory, timesheets, project, purchase, helpdesk, documents*
5. Aplikasi human resources : *recruitment, employees, fleet, leaves, appraisal.*
6. Aplikasi marketing : *marketing automation, mass mailing, events, survey.*
7. Aplikasi manufaktur : *MRP, PLM, equipment, quality.*
8. Aplikasi komunikasi : *discuss, e-Signature.*

2.12 Metode *Prototype*

Prototype digunakan untuk menggali kebutuhan secara lebih cepat. Biasanya saat pembuatan *ptototype*, keterlibatan user sangat dibutuhkan. Manfaat utama *prototype* adalah untuk mengurangi resiko tidak diterimanya hasil pengembangan suatu perangkat lunak serta pengulangan kerja di kemudian hari, (Munawar, 2018).

Manfaat itu diantaranya :

1. Lebih antusiasnya pengguna akhir dan pelanggan dalam proses penggalian kebutuhan dan umpan balik.
2. Mengurangi resiko proyek karena penggunaan *user interface* sudah dieksplorasi dari awal.
3. Kemudahan dalam penggunaan dapat lebih ditingkatkan.
4. Bisa menjembatani antara produk perangkat lunak dengan kebutuhan *user*.



Gambar 2.1 Metode *Prototype*

Langkah-langkah dalam metode *Prototype* dimulai dengan dilakukannya Komunikasi antara Tim Pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para *Stakeholder* untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun yang saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada literasi selanjutnya merupakan keharusan. Literasi pembuatan *Prototype* direncanakan dengan cepat dan pemodelan (dalam bentuk “Rancangan Cepat”) dilakukan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan terlihat oleh para pengguna

akhir (Misalnya rancangan antarmuka pengguna *User* atau format tampilan). Rancangan Cepat (*Quick Design*) akan memulai konstruksi pembuatan *Prototype*. Lalu kemudian akan diserahkan kepada para *stakeholder* dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap *Prototype* yang telah dibuat sebelumnya, kemudian akan memberikan umpan balik, (Roger S. Pressman, 2012).

2.13 *Unified Modelling Language (UML)*

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*. Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan interatif, yaitu : identifikasi kelas-kelas dan objek-objek, identifikasi semantic dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. UML dibangun atas model 4+1 *view*. Yaitu *LogicalView*, *Development View*, *Process View*, *Physical View* dan *Scenario*. Model ini di dasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 *View*, yang salah satunya adalah *Scenario*, (Munawar, 2018).

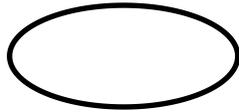
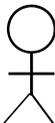
2.13.1 *Use Case Diagram*

Use Case adalah deskripsi fungsi sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut sebagai *Scenario*. Setiap *Scenario* menggambarkan urutan

kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian, secara singkat bias dikatakan *Use Case* adalah serangkaian *Scenario* yang digabungkan bersaa-sama oleh tujuan umum pengguna. *Use Case* dibuat berdasarkan kebutuhan Aktor. *Use Case* harus merupakan 'apa' yang dikerjakan *software* aplikasi, bukan 'bagaimana' *software* aplikasi mengerjakannya, (Munawar, 2018 : 89).

Tabel 2.1 berikut ini adalah Simbol-simbol yang digunakan dalam *Use Case Diagram* :

Tabel 2.1 Simbol *Use Case Diagram*

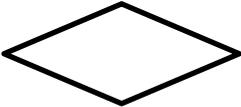
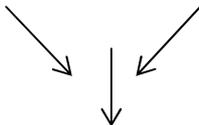
No	Simbol	Keterangan
1	Nama <i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antara unit atau <i>factor</i> , biasanya dinyatakan dengan menggunakan kata kerja awal frase nama <i>Use Case</i>
2	Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri. Jadi, walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	Asosiasi 	Komunikasi antara aktor dan <i>Use Case</i> , atau <i>Use Case</i> dan Aktor
4	Generalisasi 	Hubungan <i>Generalisasi</i> dan <i>Spesialisasi</i> (Umum - Khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5	Ekstensi / <i>Extend</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>Use Case</i> tambahan.
6	<i>Include</i>  <i>Uses</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> , dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.

2.13.2 Activity Diagram

Diagram adalah bagian penting dari *UML* yang menggambarkan aspek dinamis dari Sistem. Logika Prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah di deskripsikan dalam *Activity Diagram*. *Activity Diagram* mempunyai peran seperti halnya *Flowchart*, akan tetapi perbedaannya dengan *Flowchart* adalah *Activity Diagram* bisa mendukung perilaku paralel sedangkan *Flowchart* tidak bisa, (Munawar, 2018 : 127).

Tabel 2.2 berikut ini adalah Simbol-simbol yang digunakan dalam *Activity Diagram* :

Tabel 2.2 Simbol *Activity Diagram*

No	Simbol	Keterangan
1	Status Awal 	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
2	Aktivitas 	Aktivitas yang dilakukan suatu sistem.
3	Percabangan / <i>Decision</i> 	Simbol ini digunakan jika ada pilihan aktivitas lebih dari satu atau menggabungkan dua kegiatan paralel menjadi satu.
4	Penggabungan / <i>Join</i> 	Penggabungan / <i>Join</i> digunakan untuk menunjukkan adanya kegiatan yang digabungkan.
5	Status Akhir 	Status Akhir, adalah akhir dari aktifitas sebuah sistem.

2.14 Penelitian Terkait

Tabel 2.3 berikut ini adalah penelitian terkait yang berkaitan dengan penelitian yang dilakukan saat ini:

Tabel 2.3 Penelitian Terkait

No	Nama Penulis	Judul / Tahun Terbit	Uraian
1	Rolia Bariang, R. Fanry Siahaan	Sistem Pendukung Keputusan Dalam Menentukan Bantuan Pupuk Subsidi Kepada Kelompok Tani Menggunakan Metode <i>Technique For Others Reference By Similarity To Ideal Solution</i> (Topsis).	<i>Technique For Others Reference By Similarity To Ideal Solution</i> (TOPSIS) merupakan suatu model pendukung keputusan dalam SPK, Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki, hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif.
		JIKOMSI [Jurnal Ilmu Komputer dan Sistem Informasi] Vol.3 No.3, September-Maret 2021, pp 118-126	Penelitian dilakukan dengan metode yang berbeda yaitu metode <i>K-Nearst Neighbor</i> dimana metode yang digunakan dalam penelitian ini jauh lebih sederhana namun tetap mendapatkan hasil yang akurat.
		Pengembangan yang dilakukan	
2	Juan Prima Gultom , Alex Rikki	Implementasi <i>Data Mining</i> menggunakan Algoritma C-45 pada Data Masyarakat Kecamatan Garoga untuk Menentukan Pola Penerima Beras Raskin.	Kesenjangan taraf kehidupan masyarakat diwilayah Kecamatan Garoga sampai saat ini belum merata, masih banyak masyarakat miskin. Program Bantuan sosial salah satu komponen program

		KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer - Vol. 02 No. 01 (April 2020))	jaminan sosial yang menjadi bentuk ekspresi tanggung jawab pemerintah, pemerintah daerah yang sangat peduli terhadap kondisi masyarakat yang miskin dan terlantar. Algoritma C4.5 merupakan algoritma klasifikasi pohon keputusan yang banyak digunakan karena memiliki kelebihan utama dari algoritma yang lainnya. Tujuan dari penelitian ini untuk mengetahui faktor mana saja yang paling dominan mempengaruhi data masyarakat kecamatan garoga untuk menentukan pola penerima beras raskin dengan tepat dan akurat, sehingga ini menjadi pedoman bagi pihak masyarakat ke depan untuk meningkatkan penerimaan beras raskin bagi masyarakat.
	Pengembangan yang dilakukan		Penggunaan metode yang berbeda dengan tujuan yang sama, namun penelitian ini jauh lebih simpel dalam hal pengujian karena sistem sudah berbasis Android dan dapat diakses dimana saja.
3	Riyan Latifahul Hasanah, Muhamad Hasan, Witriana Endah Pangesti,Fanny Fatma Wati, Windu Gata	Klasifikasi Penerima Dana Bantuan Desa Menggunakan Metode Knn (<i>K-Nearest Neighbor</i>). Jurnal TECHNO Nusa Mandiri Vol. 16, No. 1 Maret 2019	Penentuan status keluarga miskin sebagai penerima bantuan merupakan hal yang sangat penting agar bantuan penanggulangan kemiskinan dari pemerintah dapat disalurkan secara tepat sasaran. <i>Data mining</i> memanfaatkan pengalaman atau bahkan kesalahan di masa lalu untuk meningkatkan kualitas dari model maupun hasil analisisnya, salah satunya dengan kemampuan yang dimiliki teknik <i>data mining</i> yaitu klasifikasi. Dalam <i>dataset</i>

			penerima bantuan yang digunakan dalam penelitian ini, terdapat 159 record atau tuple dengan empat atribut (kondisi rumah, penghasilan, pekerjaan dan jumlah tanggungan). Prediksi kategori data baru dilakukan dengan menggunakan tahapan perhitungan manual <i>Euclidean Distance</i> dari lima nilai K yang berbeda.
	Pengembangan yang dilakukan		Penelitian ini menggunakan tahapan perhitungan otomatis yang sudah diimplementasikan kedalam bentuk aplikasi berbasis Android.
4	Tia Noviana , Jasmir, Yudi Novianto	Penerapan <i>Data Mining</i> Menentukan Kelompok Prioritas Penerima Bantuan Beras Rastra Dengan <i>Clustering K-Means</i> . Jurnal Ilmiah Mahasiswa Teknik Informatika, [S.l.], v. 1, n. 3, p. 159-174, dec. 2019.	Setiap Kecamatan di Provinsi Jambi mendapatkan kesempatan untuk mengelola bantuan Beras Sejahtera (rastra). Selain tingkat akurasi data juga dibutuhkan efisiensi waktu pengolahan data penerima bantuan. Terdapat kerumitan dalam pengolahan data selama ini, yaitu menentukan penduduk yang menjadi prioritas utama untuk mendapatkan bantuan ditengah banyaknya data penduduk di Kecamatan Muara Papalik. Dalam melakukan analisis, penulis menggunakan data penduduk dari tahun 2017 sampai tahun 2018 kemudian disajikan kedalam format Arff. Dalam melakukan analisis, penulis menggunakan alat bantu tools WEKA dengan metode <i>Clustering K-means</i>
	Pengembangan yang dilakukan		Penelitian yang dilakukan menggunakan metode yang berbeda, pemanfaatan android menjadi salah satu pengembangan yang

			dilakukan.
5	Feven Indriyani	<p>Rancang Bangun Aplikasi Monitoring Hasil Produksi dan Informasi Preservasi (Perawatan) Tanaman Karet Menggunakan Metode K-Nearest Neighbor.</p> <p>Repo IIB DARMAJAYA (2019). http://repo.darmajaya.ac.id/id/eprint/1435</p>	<p>Suatu proses yang dilakukan secara terus menerus yang merupakan bagian yang bersifat integral dari manajemen yang meliputi penilaian yang bersifat sistematis terhadap kemajuan suatu pekerjaan. Monitoring adalah penilaian yang sistematis dan terus menerus terhadap kemajuan suatu pekerjaan. Aplikasi monitoring hasil produksi PT. Perkebunan Nusantara Vii Unit Muara Enim merupakan salah satu bentuk sarana aplikasi yang bertujuan mempermudah mandor dalam penginputan hasil produksi dan mempermudah asisten kepala wilayah dalam memonitoring hasil produksi berdasarkan hasil produksi yang diinputkan, serta tersedia informasi tentang preservasi tanaman karet.</p>
		Pengembangan yang dilakukan	<p>Penelitian sebelumnya menggunakan metode yang sama namun dengan tujuan yang berbeda. Pemanfaatan basis Android menjadi salah satu pengembangan yang dilakukan pada penelitian ini.</p>