PENERAPAN OUTLIER ANALYSIS SEBAGAI SALAH SATU REKOMENDASI KELOMPOK BELAJAR TERHADAP SISWA KELAS 6 DI SDN PAGELARAN II

Idham Nugraha[1], Dian Dharmayanti[2]
[1,2] Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-114 Bandung
E-mail : idhamnugraha@gmail.com[1], dian.dharmayanti@email.unikom.com[1]

ABSTRAK

SDN Pagelaran II merupakan salah satu Sekolah Dasar di Kabupaten Cianjur yang berlokasi di Kampung Sinagar, Desa Pagelaran, Kecamatan Pagelaran. Untuk dapat dinyatakan lulus siswa harus bisa melewati batas Kriteria Ketuntasan Minimun (KKM) yang telah ditentukan pada masing-masing mata pelajaran. Selain kelas tambahan dan *try out* bagi kelas 6, biasanya dilakukan juga pembentukan kelompok belajar untuk setiap mata pelajaran. Permasalahannya adalah pihak sekolah biasanya membagi kelompok hanya berdasarkan urutan absensi saja. Sehingga akan mengakibatkan siswa yang unggul berada dalam satu kelompok dengan siswa yang tertinggal dalam suatu mata pelajaran dan dikhawatirkan siswa unggul tersebut akan merasa bosan karena materi yang diberikan sudah dipahaminya diulang-ulang agar siswa yang tertinggal dapat mengejar ketertinggalannya.

Berdasarkan pemaparan masalah sebelumnya, solusi yang ditawarkan untuk mengatasi permasalahan tersebut adalah mencari data pencilan (*outlier*) dan membentuk pengelompokan siswa berdasarkan nilai siswa dari semester 7-10. Pencarian data pencilan dapat dilakukan dengan metode *Distance Based*, dan untuk pembentukan kelompok dapat menggunakan metode *clustering* di *data mining*. Salah satu algoritma untuk mencari data pencilan adalah algoritma *Local Outlier Factor (LOF)*, dan algoritma yang dapat digunakan untuk melakukan clustering (pengelompokan) adalah algoritma *Agglomerative Hierarchical Clustering (AHC)*.

Kata kunci : data pencilan, pengelompokan, *local outlier factor, agglomerative hierarchical clustering, data mining.*

1. PENDAHULUAN

Pada bagian ini akan dipaparkan mengenai latar belakang masalah, maksud, tujuan dan teori yang digunakan.

1.1. Latar Belakang Masalah

Bagi seorang siswa, belajar merupakan suatu kewajiban. Berhasil atau tidaknya seorang siswa dalam pendidikan tergantung pada proses belajar yang dialami oleh siswa tersebut. Disetiap tingkat kelas yang dilalui siswa harus melalui ujian, mulai dari ulangan harian, ujian tengah semester (UTS) dan ujian akhir semester (UAS). SDN Pagelaran II adalah salah satu Sekolah Dasar di Kabupaten Cianjur, tepatnya di Kp. Sinagar Desa Pagelaran Kecamatan Pagelaran. Sebagai sekolah yang menjadi salah satu pilihan, SDN Pagelaran II harus meningkatkan kualitas pendidikannya.

Berdasarkan hasil observasi dan wawancara yang dilakukan dengan pihak guru dan pihak kurikulum di SDN Pagelaran II, pemberian materi yang diajarkan dilakukan dengan metode yang sama tanpa membedakan siswa satu dengan yang lainnya. Setelah dilakukan ulangan harian, tetap saja ada siswa yang mendapat nilai lebih kecil dari rata-rata nilai kelas dalam mata pelajaran dan ada siswa yang selalu mendapat nilai tinggi dalam mata pelajaran tertentu, karena setiap siswa memiliki potensi yang berbeda satu sama lain. Untuk dapat dinyatakan lulus siswa harus bisa melewati batas Kriteria Ketuntasan Minimun (KKM) yang telah ditentukan pada masing-masing mata pelajaran. Selain kelas tambahan dan try out bagi kelas 6, biasanya dilakukan juga pembentukan kelompok belajar untuk setiap mata pelajaran di setiap tingkat kelas, untuk membentuk kelompok belajar biasanya guru akan membagi siswa berdasarkan urutan absen atau siswa yang memilih teman kelompoknya sendiri. Hal ini memungkinkan siswa yang meiliki nilai tinggi akan berada dalam satu kelompok dengan siswa yang tertinggal di mata pelajaran tersebut. Hal inilah yang menyebabkan siswa dengan nilai tinggi akan merasa bosan karena materi yang sudah dipahaminya diulang-ulang untuk membantu siswa dengan nilai rendah untuk memahami materi tersebut.

Agar siswa dapat mengejar mata pelajaran yang tertinggal dan dapat mencapai KKM, kelompok belajar harus dibentuk sesuai dengan kemiripan nilai siswa. Sehingga akan mempermudah guru untuk mengidentifikasi dimanakah kesulitan yang dialami oleh siswa dalam kelompok yang terbentuk untuk memahami materi pelajaran yang telah diberikan. Untuk mendapatkan informasi tersebut secara lebih cepat dan efisien perlu kiranya suatu bantuan teknologi informasi, dalam hal ini yaitu data mining atau metode yang digunakan untuk melihat pola atau

set data yang berukuran besar [1]. Dalam data mining terdapat beberapa metode yang dapat digunakan tergantung dari tujuan yang akan dicapai. Salah satu metode yang dapat digunakan adalah metode outlier analisis. Metode Outlier analysis atau Deteksi Anomali adalah himpunan data yang sangat berbeda dengan data lainnya [2], sedangkan metode clustering adalah suatu teknik dalam data mining untuk mengelompokan data yang memiliki kemiripan karakteristik dengan data lainnya. Dengan menggunakan metode clustering maka kelompok belajar yang dihasilkan akan beranggotakan siswa-siswa yang memiliki kemiripan dalam nilai akademik suatu mata pelajaran dan akan ditemukan outlier dari setiap kelompok yang terbentuk.

Sehingga untuk bisa memetakan siswa yang mendapat nilai rendah pada setiap kelompok-kelompok belajar yang sesuai adalah dengan outlier analysis pada tahapan cluster.

1.2. Maksud dan Tujuan

1.2.1. Maksud

Maksud dari penelitian yang akan dilakukan adalah penerapan *outlier analysis* di SDN Pagelaran II dalam pembentukan kelompok belajar.

1.2.2. Tujuan

Sedangkan tujuan yang akan dicapai dalam penelitian ini yaitu untuk membantu pihak kurikulum :

1. Menemukan data outlier yang akan menjadi kelompok belajar siswa.
2. Membentuk kelompok belajar berdasarkan kemiripan nilai siswa.

1.3. Data

Data adalah suatu nilai yang merepresentasikan fakta di dunia nyata dan mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dsb. Nilai tersebut lalu direkam dalam bentuk angka, huruf, simbol,teks, gambar, atau kombinasinya [4].

1.4. Basis Data

Basis data adalah sekumpulan data yang saling terhubung dan saling mempunyai skema tertentu.

Menurut Fathansyah [1], basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

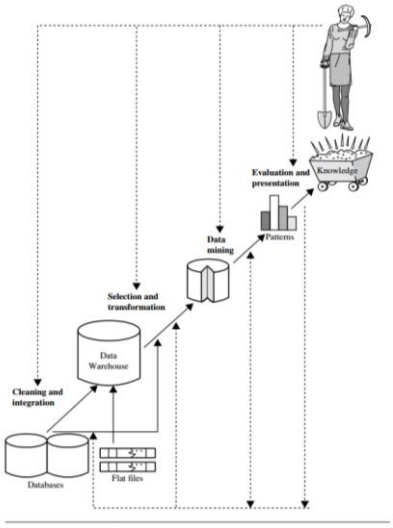
Kumpulan *file* atau table atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

1.5. *Data Mining*

Data mining adalah kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar. Nama lain yang sering digunakan dan memiliki makna yang sama dengan *data mining* adalah *knowledge mining from data*, *knowledge extraction*, *pattern analysis*, *data archaeology*, *data dredging*.

Data mining berusaha memecahkan masalah dengan menganalisis sekumpulan data yang sudah ada dalam basis data. Sedangkan hasil dari proses dari *data minig* adalah penemuan pengetahuan, informasi, pola, atau model.

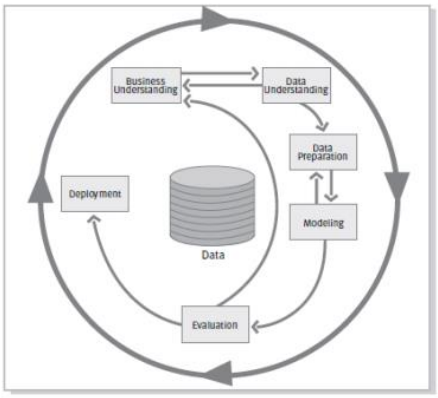
Sebagian lain melihat *data mining* hanya sebagai satu langkah penting pada proses KDD (*Knowledge Discovery from Data*). Gambar merupakan langkah-langkah dari proses KDD [2].



Gambar 1. Tahapan KDD

1.6. Metode CRISP-DM

Dalam CRISP-DM, sebuah proyek data mining memiliki siklus hidup yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah [3]. Berikut adalah fase yang ada dalam siklus CRISP-DM, yaitu :



Gambar 2. Proses Data Mining menurut CRISP-DM

1.7. *Outlier Analysis*

Outlier berbeda dari data *noise*. Noise adalah *random error atau varians* dalam variabel yang diukur. Secara umum, *noise* ini tidak menarik dalam analisis data, termasuk deteksi *outlier*. Sebagai contoh, dalam deteksi penipuan kartu kredit, perilaku pembelian pelanggan dapat dimodelkan sebagai sebuah variabel acak. Sebagai contoh pelanggan dapat menghasilkan beberapa "*noise* transaksi" yang mungkin tampak seperti "kesalahan yang acak" atau "*varians*," seperti dengan membeli makan siang yang lebih besar satu hari, atau memiliki satu lebih cangkir kopi dari biasanya. Transaksi tersebut tidak akan diperlakukan sebagai *outlier*; Jika tidak, perusahaan kartu kredit akan dikenakan biaya berat dari memverifikasi bahwa banyak transaksi. Perusahaan juga akan kehilangan pelanggan oleh mengganggu mereka dengan beberapa alarm palsu. Seperti banyak lainnya analisis data dan pertambangan data tugas, *noise* harus dihapus sebelum deteksi *outlier*.

Outliers menarik karena mereka diduga tidak dihasilkan oleh mekanisme yang sama sebagai sisa data. Oleh karena itu, dalam deteksi *outlier*, penting untuk membenarkan mengapa outliers terdeteksi dihasilkan oleh beberapa mekanisme lain. Hal ini sering dicapai dengan membuat berbagai asumsi di sisa data dan menunjukkan bahwa outliers terdeteksi melanggar anggapan-anggapan secara signifikan. Deteksi *outlier* juga berkaitan dengan *novelty* deteksi dalam berkembang data set. Sebagai contoh, dengan memantau situs web media sosial mana konten baru masuk, *novelty* deteksi dapat mengidentifikasi tren dan topik baru secara tepat waktu. *Novelty* topik mungkin pada awalnya muncul sebagai *outliers*. Sejauh ini, *outlier* deteksi dan *novelty* deteksi berbagi beberapa kesamaan dalam metode Deteksi dan pemodelan. Namun, kritis perbedaan antara keduanya adalah bahwa di *novelty* deteksi, setelah topik baru dikonfirmasi, mereka biasanya dimasukkan ke dalam model perilaku normal sehingga kasus tindak lanjut tidak diperlakukan sebagai *outliers* lagi [2].

1. Metode Pendeteksian *Outlier*

Berikut adalah beberapa metode untuk membuat model pendeteksian *outlier*, yaitu *Supervised, Semi-Supervised* dan *Unsuported Methods*.

a. *Supervised Methods*
   Supervised method model data normal dan abnormal. Para pakar domain memeriksa dan label sampel data yang mendasarinya. *Supervised method* dapat kemudian dimodelkan sebagai klasifikasi masalah. Tugas kita adalah untuk mempelajari classifier yang dapat mengenali *outlier*. Sampel digunakan untuk pelatihan dan pengujian.
b. *Semi- Supervised Methhods*
   Semi-supervised method outlier dapat dianggap sebagai aplikasi semi diawasi pembelajaran. Sebagai contoh, ketika beberapa objek normal berlabel tersedia, kita dapat menggunakan mereka, bersama dengan benda-benda tanpa label yang berada di dekatnya, untuk melatih sebuah model untuk objek biasa. Model objek biasa kemudian dapat digunakan untuk mendeteksi objek *outlier* tersebut tidak cocok model objek biasa diklasifikasikan sebagai *outlier*.
c. *Unsupervised Methods*
   Unsupervised method outlier membuat sebuah asumsi implisit: objek normal yang bias disebut sebagai "*cluster*". Dengan kata lain, *Unsupervised method outlier* mengharapkan bahwa objek biasa mengikuti pola yang jauh lebih sering daripada *outlier*. Objek biasa tidak memiliki jatuh ke dalam satu kelompok dengan kesamaan yang tinggi. Sebaliknya, mereka dapat membentuk beberapa kelompok, dimana setiap kelompok memiliki fitur yang berbeda. Namun, outlier diharapkan terjadi jauh di ruang fitur dari salah satu kelompok-kelompok objek biasa.

Dalam method pendeteksian *outlier* membuat asumsi mengenai *outlier* dengan data yang ada. Maka dari asumsi yang tercipta kita dapat mengkategorikan metode pendeteksian *outlier* pada tiga tipe, yaitu *Statistical Method, Proximity-Based Methods*, dan *Clustering-Based Methods* [2].

a. *Statistical Method*
   Statistical method (juga dikenal sebagai model-berdasarkan metode) membuat asumsi normalitas data. Mereka menganggap bahwa objek normal data yang dihasilkan oleh model (stokastik) statistik, dan data yang tidak mengikuti model yang outliers.
b. *Proximity-Based Methods*
   Proximity-bsed methods berasumsi bahwa objek merupakan *outlier* jika tetangga terdekat objek jauh di fitur ruang, yaitu kedekatan objek tetangganya secara signifikan menyimpang dari kedekatan

sebagian besar benda-benda lain untuk tetangga mereka set data yang sama.

c. *Clustering-Based Methods*
Clustering-based method berasumsi bahwa objek normal data memiliki kelompok data yang besar dan padat, sedangkan *outlier* hanya memiliki kelompok kecil atau tipis, atau tidak termasuk kedalam setiap *cluster*.

2. *Statistical Approaches*
Metode pendeteksian *outlier* Statistik (atau metode berbasis model) menganggap bahwa objek normal data mengikuti model statistik, dimana data tidak mengikuti model dianggap *outliers*. Metode tersebut mungkin *parametric* atau *nonparametric* [2].

3. *Proximity-Based Approaches*
Metode pendeteksian berbasis kedekatan outlier menganggap bahwa objek *outlier*.Jika kedekatan objek dengan tetangganya yang terdekat secara signifikan menyimpang dari kedekatan sebagian besar benda-benda lain untuk tetangga mereka set data yang sama. Metode pendeteksian berbasis jarak *outlier* dengan lingkungan suatu objek, yang didefinisikan oleh radius tertentu. Objek adalah *outlier* jika lingkungan yang tidak memiliki cukup poin lain. Dalam metode deteksi *outlier* berbasis kepadatan, objek merupakan *outlier* Jika rapatan energinya relatif jauh lebih rendah dibandingkan negara tetangga [2]. Salah satu algoritma yang dapat digunakan adalah *Local Outlier Factor* (LOF).
Algoritma LOF :
Ide utama dari LOF adalah membandingkan kepadatan lokal lingkungan sebuah objek dengan kepadatan lokal tetangganya, dalam algoritma LOF sebuah objek dikatakan sebagai *outlier* apabila memiliki nilai LOF yang tinggi atau menjauhi 1, sedangkan objek yang memiliki nilai LOF rendah atau mendekati 1 maka objek tersebut tidak dapat dikategorikan sebagai outlier. Nilai LOF yang tinggi mengindikasikan bahwa objek tersebut memiliki kepadatan yang rendah terhadap lingkungannya, sehingga berpotensi menjadi outlier. Langkah-langkah untuk menemukan outlier di deskripsikan sebagai berikut :

a. Menghitung *k-distance* dari setiap objek p.
Tujuan dari perhitungan *k-distance* ini adalah untuk menentukan tetangga dari p, secara sederhana *k-distance* dari sebuah objek p adalah jarak maksimal dari p terhadap tetangga terdekatnya dan dinotasikan dengan *k-distance(p), k-distance* didefinisikan d(p,o) dimana antara p dan objek o ∈ D memiliki :
   (i) Untuk setidaknya *k* objek o' ∈ D | {p} dan dinyatakan bahwa $d(p,o') \leq d((p,o)$
   (ii) Untuk setiap *k*-1 objek o' ∈ D | {p} dan dinyatakan bahwa $d(p,o') \leq d((p,o)$

b. Menghitung tetangga terdekat yang tidak lebih dari nilai *k-distance.*
k-deistance neighborhood suatu objek p dinotasikan *Nk-distance(p),* atau *Nk(p)* dimana berisi setiap objek dengan jarak lebih besar dari *k-distance(p).*

c. Menghitung *Reachability Distance* dari p, mencari jarak maksimum antar objek.
Reachability Distance dari suatu objek p terhadap objek o adalah *distance(p, o)* atau *k-distance(o),* dengan membandingkan keduanya dan dicari nilai yang maksimum sehingga $reach\text{-}dist_k(p,o) = max\{k\text{-}distance\ (o), distance(po)\}$. Tujuan dari perhitungan ini adalah untuk memastikan bahwa semua benda berada dalam lingkungan yang homogen. Selain itu, nilai LOF akan stabil jika sebuah objek berada dalam lingkungan yang seragam sekalipun jika *MinPts(k)* berubah.
Fluktuasi dari *reachability distance* dapat dikontrol dengan memilih nilai maksimum untuk k.

d. Menghitung *local reachability density* (kepadatan lokal) dari setiap objek. Dengan persamaan berikut.

$$\boldsymbol{lrd_{MinPts}(p)} = \frac{\boldsymbol{o} \in \sum_{N\ MinPts} \boldsymbol{reachdist_{MinPts}(p,o)}}{|\boldsymbol{N_{MinPts}(p)}|}$$

Keterangan :
$lrd_{MinPts}(o)$: *local reachability density* dari objek p.
reachdist(p) : *Reacbility distance* dari objek p ke objek o
$N_{MinPts}(p)$: Jumlah tetangga p dalam suatu minPts

Secara intuitif, *local reachbility density* p adalah kebalikan dari rata-rata *reachability distance based* didalam minPts-tetangga terdekat dari objek p.

e. Menghitung LOF untuk setiap objek data dengan persamaan berikut.

$$\boldsymbol{LOF_{MinPts}(p)} = \frac{\boldsymbol{o} \in \sum_{N\ MinPts} \frac{\boldsymbol{lrdMinPts(o)}}{\boldsymbol{lrdMinPts(p)}}}{|\boldsymbol{N_{MinPts}(p)}|}$$

Keterangan :
$LOF_{MinPts}(p)$: Derajat outlier dari objek p
$lrd_{MinPts}(o)$: *local reachabiliry distance* dari objek p.
reachdist(p) : *Reacbility distance* dari objek p ke objek o
$N_{MinPts}(p)$: Jumlah tetangga p dalam suatu minPts

LOF adalah sebuah derajat yang menentukan apakah sebuat objek merupakan *outlier* atau tidak. LOF dinotasikan kedalam $LOF_{MinPts}^{(p)}$

dan merupakan rata-rata rasio kepadatan *reachability* lokal p dan tetangga p dalam satu jangkauan.

1.8. *Clustering*

Clustering adalah proses mengelompokkan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas atau cluster [2]. Tujuannya menemukan *cluster* yang berkualitas dalam waktu yang layak. Clustering dalam data mining berguna untuk menemukan pola distribusi di dalam sebuah data set yang berguna untuk proses analisa data. Kesamaan objek biasanya diperoleh dari kedekatan nilai-nilai atribut yang menjelaskan objek-objek data, sedangkan objek-objek data biasanya direpresentasikan sebagai sebuah titik dalam ruang multidimensi.

Dengan menggunakan *clustering*, dapat diidentifikasi daerah yang padat, pola-pola distribusi secara keseluruhan dan keterkaitan yang menarik antara atribut-atribut data. Dalam data mining usaha difokuskan pada metode-metode penemuan untuk cluster pada basis data berukuran besar secara efektif dan efisien.

1.9. *Agglomerative Hierarchical Clustering*

Pengelompokan data menggunakan *agglomerative hierarchical clustering* (AHC) diawali dengan merepresentasikan setiap objek pada data sebagai satu kelompok, kemudian dilakukan perhitungan jarak (*distance measure*) antar kelompok tersebut. Setelah itu dua kelompok yang memiliki jarak terdekat digabungkan menjadi sebuah kelompok baru. Kemudian jarak antara kelompok yang baru dengan kelompok lain dihitung menggunakan salah satu metode perhitungan jarak antar kelompok (*single linkage, complete linkage, average linkage,* dll.) [4]. Proses perhitungan jarak dan penggabungan dua kelompok dilakukan secara iteratif. Berikut ini merupakan beberapa metode yang dapat digunakan untuk menghitung jarak antar kelompok :

a. *Single Linkage* (Jarak Terdekat)
   Single linkage disebut juga minimum *link*, di mana *similaritas* dari dua *cluster* didasarkan pada dua titik paling dekat dari dua *cluster* yang berbeda. Untuk mencari *single linkage* dapat menggunakan persamaan (2.8) berikut ini :
   $$\boldsymbol{d_{uv} = \min\{d_{uv}\}, d_{uv} \in D}$$
b. *Complete Linkage* (Jarak Terjauh)
   Complete linkage disebut juga *maximum link*, dimana jarak antara dua buah *cluster* dihitung dari jarak terjauh antara anggota *cluster* satu dengan *cluster* kedua. Untuk mencari *complete linkage* dapat menggunakan persamaan (2.9) berikut ini :
   $$\boldsymbol{d_{uv} = \max\{d_{uv}\}, d_{uv} \in D}$$
c. *Average Linkage* (Jarak Rata-rata)
   Average linkage disebut juga *group average*, di mana jarak antara dua buah *cluster* dihitung dari rata-rata jarak antara anggota *cluster* satu dengan *cluster* yang kedua. Untuk mencari *average linkage* dapat menggunakan persamaan (2.10) berikut ini :
   $$\boldsymbol{d_{uv} = \text{average}\{d_{uv}\}, d_{uv} \in D}$$

Berikut ini merupakan langkah-langkah dalam algoritma *Agglomerative Hierarchical Clustering* (AHC) :

1. Setiap objek data dianggap sebagai *cluster*. Jika n = jumlah data dan k = jumlah *cluster*, maka k = n.
2. Menghitung matriks jarak antar *cluster* (menggunakan *euclidean distance*)
3. Cari dua *cluster* yang mempunyai jarak paling dekat (*single linkage*) lalu gabungkan dua *cluster* tersebut, sehingga k = k-1.
4. Jika k > 1, ulangi langkah 2 dan 3
5. Jika k = 1, perulangan selesai.

1.10. *Usability Testing*

Pengujian *usability* dapat dilakukan dengan melibatkan pengguna atau tanpa melibatkan pengguna. Pengujian dengan melibatkan pengguna dapat memberikan informasi langsung dari pengguna tentang bagaimana pengguna menggunakan sistem serta permasalahan yang dihadapi. Pada metode observasi langsung, seorang observator mengamati satu atau lebih pengguna di lokasi. Sebaiknya seorang observator tidak memperlihatkan diri agar pengujian sistem berjalan normal. Untuk membantu proses observasi dapat menggunakan video kamera yang merekam ekspresi pengguna. Metode kuesioner dilakukan dengan menggunakan kuesioner untuk mengukur kepuasan pengguna dan untuk mengetahui opini pengguna terhadap aplikasi yang digunakan. Metode Thinking Aloud dilakukan dengan mengamati secara terus menerus sikap tubuh dan ucapan yang ditunjukkan oleh seorang responden pada saat menggunakan aplikasi [5].

Tingkat efektivitas dan efisiensi diukur dengan menggunakan *user's success rate* (tingkat keberhasilan pengguna) yaitu persentase tugas yang diselesaikan dengan benar oleh pengguna. Efektivitas dan efisiensi dihitung menggunakan persamaan berikut :

$$\textbf{Efektifitas, efisiensi } (\%) = \frac{(\sum_{i=1}^{n} Xi)}{\mathbf{n}} \textbf{ x 100\%}$$

Keterangan :

Xi : nilai keberhasilan pengguna, Xi = {0,1}

Kepuasan adalah persentase perbandingan antara nilai kepuasan pengguna dengan perkalian antara bobot maksimum skala Likert dengan jumlah responden

$$\textbf{Kepuasan } (\%) = \frac{(\sum_{i=1}^{n} Xi)}{\mathbf{5 \text{ x } n}} \textbf{ x 100\%}$$

Keterangan :

Xi : nilai keberhasilan pengguna, Xi = {0, 1, 2, 3, 4, 5}

Usability aplikasi adalah rataan dari efektivitas, efisiensi dan kepuasan, seperti tertulis pada persamaan berikut :

$$\textbf{\textit{Usability}}(\%) = \frac{(\textbf{Efektivitas+Efisiensi+Kepuasan})}{\textbf{3}} \textbf{x100\%}$$

2. ISI PENELITIAN

Berikut adalah analisis dan pengujian yang dilakukan dalam penelitian

2.1. Analisis Masalah

Berdasarkan hasil observasi dan wawancara dengan pihak guru dan pihak kurikulum SDN Pagelaran II, didapatkan hasil yaitu terdapat siswa dengan nilai rendah atau kurang dari kriteria ketuntasan minimum (KKM) dari mata pelajaran tertentu.

2.2. Pemahaman Bisnis

Tahap pemahaman bisnis adalah proses awal pada kerangka kerja CRISP-DM. tahap ini berfokus untuk memahami tujuan dan kebutuhan dari sudut pandang bisnis SDN Pagelaran II. Terdapat beberapa tugas pada tahap ini, antara lain :

2.2.1. Identifikasi Tujuan Bisnis

Tujuan bisnis dari SDN pagelaran II adalah untuk melakukan proses pembelajaran dan membimbing seluruh siswa, terutama siswa yang selalu mendapat nilai rendah di beberapa mata pelajaran agar dapat lulus pada setiap mata pelajaran dengan nilai yang baik, apakah dengan cara metode pembelajaran secara langsung / pendekatan pembelajaran yang berorientasi kepada guru, pembelajaran dengan diskusi, pembelajaran dengan kelompok kecil, atau dengan membantu siswa agar dapat memahami dan menguasai materi pembelajaran strategi pemecahan masalah.

2.2.2. Identifikasi Penilaian Situasi Bisnis

Penilaian situasi bisnis yang sedang berjalan di SDN Pagelaran II meliputi daftar sumber daya komputasi (*hardware),* dan sumber daya manusia (*user)* yang ada di SDN Pagelaran II.

2.2.3. Penentuan Sasaran Data Mining

Penentuan sasaran *data mining* terdiri dari tujuan dan kriteria sukses *data mining*. Berikut ini merupakan tujuan dan kriteria sukses *data mining*.

1. Tujuan *data mining*
   Tujuan dari penerapan *data mining* dalam penelitian ini yaitu untuk menemukan *outlier*, dan membentuk kelompok belajar berdasarkan nilai dari mata pelajaran yang termasuk keladalam Ujian Nasional Sekolah Dasar pada siswa kelas 6 tahun pelajaran 2016/2017, dari semester 7 hingga semester 10 untuk yang memiliki karakteristik yang sama.
2. Kriteria sukses *data mining*
   Kriteria sukses dalam penelitian ini yaitu apabila mampu membantu pihak guru dan pihak kurikulum di SDN Pagelaran II dalam menemukan siswa yang termasuk ke dalam *outlier*.

2.2.4. Menentukan Rencana Proyek

Rencana proyek dari penelitian ini adalah dapat menerapkan analisis *outlier* pada nilai siswa di SDN Pagelaran II dengan teknik *Density-Based* dan algoritma *Local Outlier Factor (LOF)*, dan untuk membentuk *cluster* digunakan algoritma *Agglomeration Hierarchical Clustering (AHC).*

2.3. Pemahaman Data

Tahap pemahaman data merupakan tahapan kedua yang di dalamnya terdapat beberapa tugas, yaitu :

2.3.1. Pengumpulan Data Awal

Data awal yang telah dikumpulkan adalah data nilai raport siswa kelas 6 tahun 2016/2017, untuk mata pelajaran yang termasuk kedalam ujian nasional yaitu pelajaran Bahasa Indonesia, Matematika, dan Ilmu Pengetahuan Alam dari tahun pelajaran 2014/2015 dan 2015/2016 atau nilai sejak semester 7. Data nilai siswa yang akan digunakan tersebut berupa *file* yang memiliki format *.xls atau *.xlsx.

2.3.2. Penjelasan Data

Data awal yang telah dikumpulkan akan dijelaskan pada tahap ini.

2.3.3. Eksplorasi Data

Tahapan eksplorasi data dapat membantu tujuan dari *data mining*, dalam penelitian ini eksplorasi data meliputi analisis deskriptif dan visualisasi data. Data yang akan digunakan untuk tahap ekplorasi data ini adalah data nilai siswa kelas 6 tahun pelajaran 2016/2017 untuk mata pelajaran matematika, IPA, dan Bahasa Indonesia.

2.3.4. Evaluasi Kualitas Data

Evaluasi kualitas data dilakukan terhadap atribut nilai mata pelajaran Matematika, IPA, dan Bahasa Indonesia dari semester 7 hingga semester 10. Berikut ini merupakan hasil evaluasi dari atribut-atribut tersebut :

1. *Missing value* mungkin saja ditemukan pada data nilai, tetapi pada data nilai siswa kelas 6 tahun pelajaran 2016/2017 untuk nilai Matemaika, IPA, dan Bahasa Indonesia dari semester 7 hingga semester 10 yang terdapat pada lampiran F, tidak terdapat *missing value*.
2. Berdasarkan data nilai siswa kelas kelas 6 tahun pelajaran 2016/2017 untuk nilai Matemaika, IPA, dan Bahasa Indonesia dari semester 7 hingga semester 10 akan dilihat apakah terdapat outlier atau tidak, berikut ini merupakan cara

yang dilakukan untuk menemukan data *outlier* yaitu menggunakan metode *Local Outlier Factor:*

1. Hitung jarak *k-distance* dari setiap objek p atau dari setiap data.
   Untuk menemukan nilai *k-distance* diperlukan beberapa tahapan seperti berikut ini :

a. Berdasarkan nilai siswa, kemudian hitung jarak dari setiap objek dengan menggunakan fungsi jarak *euclidian distance*. Dalam tahap ini masing – masing objek akan dihitung jarak dengan dirinya sendiri dan jarak dengan objek lainnya. Berikut ini merupakan contoh proses perhitungan jarak antar objek dengan menggunakan persamaan e*uclidean distance.*

$$d(1,1) = \sqrt{\begin{matrix}(85-85)^2+(80-80)^2+(85-85)^2+\\(75-75)^2+(89-89)^2+(85-85)^2+\\(76-76)^2+(80-80)^2+(75-75)^2+\\(82-82)^2+(75-75)^2+(76-76)^2\end{matrix}} = 0$$

b. Tentukan MinPts

MinPts adalah jumlah minimal data sehingga sebuah *cluster* dapat terbentuk. Dari beberapa kali percobaan yang dilakukan terhadap data, dengan asumsi nilai *MinPts* = 9, sehingga berdasarkan Matriks Jarak Awal, dipilih 9 jarak terdekat pertama dari sebuah objek kemudian dari hasil pencarian 9 jarak terdekat akan dicari nilai *K-Distance* yang merupakan jarak terjauh dari 9 jarak terdekat yang sudah ditemukan.

Tabel 1 Tabel K-Distance

No Urut	Nomor Induk	*K-distance*	Data Dari Tetangga Ke
1	11.12.01.006	26,46	14
2	11.12.01.010	29,33	7
3	11.12.01.004	23,20	7
4	11.12.01.002	26,51	13
5	11.12.01.003	30,30	19
6	11.12.01.005	26,25	9
7	11.12.01.007	29,75	18
8	11.12.01.011	44,93	6
9	11.12.01.012	29,40	5
10	11.12.01.013	25,06	19
11	11.12.01.014	41,14	18
12	11.12.01.015	44,90	3
13	11.12.01.016	32,28	2
14	11.12.01.017	30,41	5
15	11.12.01.018	48,20	1
16	11.12.01.020	46,72	13
17	11.12.01.022	47,68	5
18	11.12.01.023	39,76	5
19	11.12.01.024	30,30	5

2. Kemudian dicari setiap tetangga yang jaraknya tidak melebihi dari *k-distance* dari objek tersebut.

Tabel 2 Tabel Data K-Distance

No Urut	Nomor induk	*k-distance neighborhood*
1	11.12.01.006	2, 4, 5, 6, 9, 10, 13, 14, 19
2	11.12.01.010	1, 3, 4, 5, 6, 7, 9, 10, 14
3	11.12.01.004	2, 4, 5, 6, 7, 9, 10, 14, 19
4	11.12.01.002	1, 3, 6, 7, 9, 10, 13, 14, 19
5	11.12.01.003	1, 2, 3, 4, 6, 7, 9, 10, 19
6	11.12.01.005	1, 2, 3, 4, 7, 9, 10, 13, 18
7	11.12.01.007	1, 2, 3, 4, 5, 6, 10, 18, 19
8	11.12.01.011	1, 2, 6, 11, 12, 13, 15, 16, 17
9	11.12.01.012	1, 2, 3, 4, 5, 6, 10, 14, 19
10	11.12.01.013	1, 3, 4, 5, 6, 7, 9, 13, 19
11	11.12.01.014	2, 6, 8, 12, 13, 15, 16, 17, 18
12	11.12.01.015	1, 2, 3, 8, 11, 13, 15, 16, 17
13	11.12.01.016	1, 3, 4, 5, 6, 7, 9, 10, 14
14	11.12.01.017	1, 2, 3, 4, 5, 6, 9, 10, 19
15	11.12.01.018	1, 2, 6, 8, 11, 12, 13, 16, 17
16	11.12.01.020	2, 6, 8, 11, 12, 13, 15, 17, 18
17	11.12.01.022	1, 2, 5, 8, 11, 12, 13, 15, 16
18	11.12.01.023	2, 3, 4, 5, 6, 7, 9, 10, 13
19	11.12.01.024	1, 2, 3, 4, 5, 7, 9, 10, 14

3. Menghitung *Reachability Distance* dari p atau dari setiap data.

Tabel 3 Data Reachability Distance

	2	4	5	6	9	10	13	14	19
1	21,977	23,07	25,44	22,52	25,68	22,38	24,06	26,46	25,63
K-Distance	29,33	26,51	30,30	26,25	29,40	25,06	32,28	30,41	30,30
Reachability Distance	29,33	26,51	30,30	26,25	29,40	25,06	32,28	30,41	30,30

4. Menghitung *local reachbility density* dari setiap objek terhadap tetangganya.

$$lrd_{MinPts}(1) = \frac{29.33 + 26.51 + 30.30 + 26.25 + 29.40 + 25.06 + 32.28 +}{9}$$

$$= \frac{259.84}{9} = 28.87$$

No Urut	Nomor induk	*Local Reachbility Density*
1	11.12.01.006	28,87
2	11.12.01.010	28,14
3	11.12.01.004	28,59
4	11.12.01.002	28,12
5	11.12.01.003	27,73
6	11.12.01.005	29,08
7	11.12.01.007	28,59
8	11.12.01.011	43,17
9	11.12.01.012	27,54
10	11.12.01.013	28,27
11	11.12.01.014	43,36
12	11.12.01.015	44,27
13	11.12.01.016	28,82
14	11.12.01.017	27,93
15	11.12.01.018	45,12
16	11.12.01.020	45,10
17	11.12.01.022	44,91
18	11.12.01.023	33,06
19	11.12.01.024	27,82

5. Setelah mendapatkan *k-distance* dan *local reachbility density*, selanjutnya adalah

menghitung *Local Outlier Factor* (*LOF*) untuk setiap objek data.

$$\boldsymbol{LOF}_{\boldsymbol{MinPts}}(\mathbf{1}) = \frac{\frac{28.87}{28.14}+\frac{28.87}{28.12}+\frac{28.87}{27.73}+\frac{28.87}{29.08}+\frac{28.87}{27.54}+\frac{28.87}{28.27}+\frac{28.87}{28.82}+\frac{28.87}{28.93}+\frac{28.87}{27.82}}{9} = 1.03$$

No Urut	Nomor induk	LOF
1	11.12.01.006	1,03
2	11.12.01.010	1,00
3	11.12.01.004	1,02
4	11.12.01.002	0,99
5	11.12.01.003	0,98
6	11.12.01.005	1,01
7	11.12.01.007	0,99
8	11.12.01.011	1,21
9	11.12.01.012	0,97
10	11.12.01.013	1,00
11	11.12.01.014	1,19
12	11.12.01.015	1,24
13	11.12.01.016	1,02
14	11.12.01.017	0,99
15	11.12.01.018	1,27
16	11.12.01.020	1,24
17	11.12.01.022	1,27
18	11.12.01.023	1,17
19	11.12.01.024	0,99

2.4. Persiapan Data

Tahapan ini mencakup semua kegiatan yang diperlukan untuk membangun data set akhir (data yang akan dimasukkan ke dalam pemodelan), terdapat beberapa tugas dalam tahap ini, yaitu :

2.4.1. Pemilihan Data

Pemilihan data memiliki tugas meliputi pemilihan atribut data yang digunakan, sehingga data tersebut dapat diproses sesuai kebutuhan *data mining*. Dalam tahap ini atribut yang digunakan adalah No Induk, nilai Matematika, IPA, dan Bahasa Indonesia (semester 7-10).

2.4.2. Pembersihan Data

Pada proses ini dilakukan proses pembersihan data, berdasarkan hasil evaluasi kualitas diperoleh bahwa pada data nilai kelas 5 untuk mata pelajaran matematika pada tahun pelajaran 2015/2016 *missing value* tidak ditemukan, tetapi jika terdapat *missing value*, maka nilai yang hilang tersebut akan diisi dengan nilai *default* 0, dan nilai yang bersifat *outlier tidak akan di proses pada tahap pembuatan kelompok dan akan menjadi kelompok tersendiri*, untuk melakukan pengelompokan siswa akan digunakan algoritma *agglomerative hierarchical clustering*, yaitu dengan melakukan proses *smoothing* terhadap data menggunakan metode *binning*.

2.4.3. Penyiapan Data Awal

Pada proses penyiapan data awal akan disiapkan data yang akan digunakan dalam tahapan pemodelan. Data yang akan digunakan dalam tahapan pemodelan adalah data nilai kelas 6 tahun pelajaran 2016/2017 dari semester 7-10.

2.4.4. Format Data

Pada proses format data adalah penentuan format data yang akan digunakan dalam pengolahan data dalam tahap pemodelan. Data yang akan digunakan dalam tahapan pemodelan adalah data nilai kelas 6 tahun pelajaran 2016/2017 dari semester 7-10 dalam bentuk *format file .xlsx atau .xls*.

2.5. Pemodelan

Pada tahap ini akan dilakukan pemodelan, pada tahap ini memiliki beberapa tugas yaitu pemilihan teknik pemodelan, penjelasan mekanisme model, serta pengujian model pada data yang telah disiapkan sebelumnya.

2.5.1. Teknik Pemodelan

Teknik pemodelan yang akan digunakan yaitu menggunakan *hierarchical clustering*, dan model yang akan digunakan untuk pengelompokkan yaitu algoritma *Agglomerative Hierarchical Clustering* (AHC).

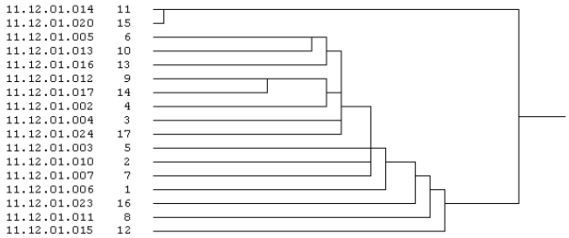
2.5.2. Analisis Pengujian AHC

Adapun tahapan dari pengujian model AHC adalah sebagai berikut :

1. Ubah setiap data menjadi *cluster*
   Inisialisasi *cluster* awal adalah sebanyak 19 *cluster* karena data(n) = *cluster*(c), kemudian dihitung jarak antara *cluster* dengan *cluster* yang ada dengan menggunakan persamaan *Euclidean Distance*.
2. Menghitung jarak antar data
   Dalam tahap ini masing – masing *cluster* akan dihitung jarak dengan dirinya sendiri dan jarak dengan *cluster* lainnya. Berikut ini merupakan proses perhitungan jarak antar *cluster* dengan menggunakan persamaan *Euclidean Distance*.

$$d(1,2) = \sqrt{\begin{matrix}(85-77)^2+(82-76)^2+(83-78)^2+\\(75-89)^2+(89-84)^2+(85-82)^2+\\(75-85)^2+(80-77)^2+(75-71)^2+\\(83-77)^2+(75-71)^2+(78-78)^2\end{matrix}} = 23.07$$

3. Mencari dua *cluster cluster* yang paling berdekatan.
4. Jika k > 1, ulangi langkah ke 2 dan 3
   Langkah ke-3 dan ke-4 terus dilakukan dan akan berhenti ketika jumlah *cluster* yang terbentuk sama dengan 1. Untuk mendapatkan 1 *cluster*, dalam kasus ini perlu dilakukan 18 iterasi.
5. Jika k = 1, perulangan selesai.
   Jumlah *cluster* akan berjumlah 1 saat perulangan ke-18.



Gambar 3 Hasil Akhir Dendogram

6. Menghasilkan *cluster* dengan memotong *dendogram* pada tingkat yang tepat berdasarkan dendogram yang telah dibentuk, tingkat yang tepat atau jumlah *cluster* terbaik akan ditentukan dengan melihat selisih jarak dari setiap *cluster* yang terbentuk. Untuk mengetahui jumlha *cluster* terbaik salah satu caranya dapat menggunakan *agglomeration schedule*. *Agglomeration Schedule* dapat dilihat pada Tabel berikut.

Tabel 4 *Agglomeration Schedule*

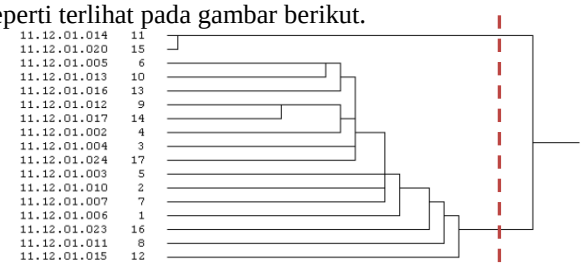
Iterasi	Kombinasi Cluster		Koefisien	Iterasi Kombinasi Cluster Sebelumnya		Iterasi Selanjutnya
	Cluster 1	Cluster 2		Cluster 1	Cluster 2	
1	11	15	0	0	0	16
2	9	14	11.057	0	0	5
3	6	10	14.629	0	0	4
4	6	13	16.125	3	0	6
5	4	9	16.348	0	2	6
6	4	6	17.521	5	4	7
7	3	4	17.671	0	6	8
8	3	17	17.896	7	0	9
9	3	5	20.839	8	0	10
10	2	3	21.172	0	9	11
11	2	7	21.448	10	0	12
12	1	2	21.977	0	11	13
13	1	16	25.08	12	0	14
14	1	8	26.87	13	0	15
15	1	12	27.713	14	0	16
16	1	11	36.073	15	1	0

Jumlah *cluster* yang terbaik dapat dilihat dari selisih koefisien dari setiap iterasinya. Selisih dari koefisien dapat dilihat pada tabel berikut.

Tabel 5 Selisih Koefisien

Jumlah Cluster	Tahap Terakhir Agglomeration	Koefisien Tahap Ini	Selisih
2	36.073	27.713	8.36
3	27.713	26.87	0.843
4	26.87	25.08	1.79
5	25.08	21.977	3.103
6	21.977	21.448	0.529
7	21.448	21.172	0.276
8	21.172	20.839	0.333
9	20.839	17.896	2.943
10	17.896	17.671	0.225
11	17.671	17.521	0.15
12	17.521	16.348	1.173
13	16.348	16.125	0.223
14	16.125	14.629	1.496
15	14.629	11.057	3.572

Selisih terbesar yaitu 8.36 sehingga *cluster* terbaik adalah 2 *cluster*. Maka, dendogram hasil pengelompokan berdasarkan ukuran akan dipotong seperti terlihat pada gambar berikut.



Gambar 4 Pemotongan *Cluster*

Sehingga kelompok yang terbentuk menjadi 3 kelompok, dengan kelompok yang terbentuk dari data *outlier*.

2.6. Pengujian

Pengujian aplikasi dilakukan terhadap fungsionalitas aplikasi untuk menilai apakah aplikasi yang dibangun sudah memenuhi kriteria atau terdapat ketidaksesuaian pada fungsionalitas sebagaimana yang terdapat pada perancangan aplikasi. Pengujian yang dilakukan untuk menguji aplikasi adalah pengujian *black box* dan *usability testing*.

2.6.1. Pengujian Black Box

Pengujian black box difokuskan pada persyaratan fungsional perangkat lunak yang dibangun.

Pengujian dilakukan dengan menguji setiap proses untuk kemungkinan kesalahan yang terjadi.

Berdasarkan hasil pengujian dengan kasus sampel uji yang telah dilakukan dapat ditarik kesimpulan bahwa seluruh fungsi berjalan dengan baik dan sesuai dengan yang diharapkan.

2.6.2. *Usability testing*

Pengujian beta merupakan pengujian yang dilakukan langsung di lingkungan sebenarnya. Pengguna memberikan penilaian terhadap aplikasi yang dibangun melalui metode *Usability Testing*. Berdasarkan hasil pengujian tersebut maka dapat ditarik kesimpulan apakah aplikasi yang dibangun sesuai dengan tujuan atau tidak. Pengujian dilakukan dengan pengguna dari aplikasi yang dibangun, yaitu Pihak Kurikulim. Pengujian ini dilakukan di SDN Pagelaran II agar dapat mengetahui sejauh mana aplikasi yang dibangun dapat menjadi solusi dan penyelesaian masalah yang sesuai dengan tujuan awal pembangunan aplikasi.

3. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan, menghasilkan kesimpulan dan saran sebagai berikut :

Kesimpulan

Hasil dari penelitian yang dilakukan diperoleh kesimpulan sebagai berikut :

1. Bahwa aplikasi yang dibangun membantu pihak bagian kurikulum untuk menemukan *outlier* dari data nilai siswa.
2. Aplikasi yang dibangun dapat membentuk kelompok belajar yang sesuai dengan kemiripan nilai siswa.

Saran

Ada beberapa saran yang dapat dilakukan untuk pengembangan aplikasi pembentukan kelompok distributor ini, antara lain :

1. Penambahan fitur pengaturan tahun ajaran agar meningkatkan keakuratan clustering dan memudahkan user untuk menyaring data hasil clustering yang akan digunakan apabila data yang dikelompokan campuran dari tahun ajaran yang berbeda.
2. Pengembangan pengelompokan dapat dikembangkan lagi dengan menggunakan algoritma lain sehingga user dapat membandingkan hasil pengelompokannya dengan algoritma *Single Linkage*.

DAFTAR PUSTAKA

[1] B. Santosa, Data Mining Teknik Pengumpulan Data Untuk Keperluan Bisnis, Yogyakarta: Andi, 2007.

[2] F. A. Hermawati, Data Mining, Surabaya: Andi, 2009.

[3] K. and E. T. Luthfi, Algoritma Data Mining, Yogyakarta: Andi, 2009.

[4] J. Han, M. Kamber and J. Pei, Data Mining Concept and Technique, Waltham: Elsevier, 2012.

[5] P. Berkhin, "Survey of Clustering Data Mining Techniques".

[6] Y. Nurhadryani, S. K. Sianturi, I. Hermadi and H. Khotimah, "Pengujian Usability Testing untuk meningkatkan Antarmuka Apliaksi Mobile," *Ilmu Komputer & Agri-Informatika*, vol. 2, pp. 83-93, 2013.

[7] Fathansyah, Basis Data, Informatika, 2015.

[8] A. Nugroho, Rekayasa Perangkat Lunak Berorientasi Objek dengan Motode USDP, Yogyakarta: Andi, 2010.

[9] R. Ramakrishnan and J. Gehrke, Database Management System Third Edition, Singapore, 2003.