KLASIFIKASI KINERJA KARYAWAN BERBASIS *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SCREEN MONITOR DURATION*

SKRIPSI

Disusun Oleh:

Muhammad Aziz Almi
H76218035

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL

SURABAYA

2022

PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini,

Nama	:	Muhammad Aziz Almi
NIM	:	H76218035
Program Studi	:	Sistem Informasi
Angkatan	:	2018

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: "KLASIFIKASI KINERJA KARYAWAN BERBASIS *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SCREEN MONITOR DURATION*". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 24 Desember 2021
Yang menyatakan,

Muhammad Aziz Almi
NIM. H76218035

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh:

NAMA : MUHAMMAD AZIZ ALMI

NIM : H76218035

JUDUL : KLASIFIKASI KINERJA KARYAWAN BERBASIS *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SCREEN MONITOR DURATION*

Ini telah diperiksa dan disetujui untuk diajukan

Surabaya, 24 Desember 2021

Dosen Pembimbing 1	Dosen Pembimbing 2
Achmad Teguh Wibowo, M.T	Mohammad Khusnu Milad, M.M.T
NIP. 198810262014031003	NIP. 197901292014031002

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Muhammad Aziz Almi ini telah dipertahankan
di depan tim penguji skripsi
Surabaya, 10 Januari 2022

Mengesahkan,
Dewan Penguji

Dosen Penguji I	Dosen Penguji II
Mujib Ridwan, S.Kom., M.T	Dr. Eng. Anang Kunaefi, M.Kom
NIP. 198604272014031004	NIP. 197911132014031001
Dosen Penguji III	Dosen Penguji IV
Achmad Teguh Wibowo, M.T	Mohammad Khusnu Milad, M.M.T
NIP. 198810262014031003	NIP. 197901292014031002

Mengetahui,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Ampel Surabaya

Prof. Dr. Hj. Evi Fatimatur Rusydiyah, M.Ag
NIP. 197312272005012003

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama	:	MUHAMMAD AZIZ ALMI
NIM	:	H76218035
Fakultas/Jurusan	:	SAINS & TEKNOLOGI / SISTEM INFORMASI
E-mail address	:	h76218035@uinsby.ac.id

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

☒ Sekripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (……)

yang berjudul :

KLASIFIKASI KINERJA KARYAWAN BERBASIS *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SCREEN MONITOR DURATION*

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 14 Januari 2021

(MUHAMMAD AZIZ ALMI)

ABSTRAK

KLASIFIKASI KINERJA KARYAWAN BERBASIS *SUPPORT VECTOR MACHINE* MENGGUNAKAN *SCREEN MONITOR DURATION*

Oleh:

Muhammad Aziz Almi

Banyaknya karyawan pada suatu perusahaan akan membuat pemilik perusahaan kesulitan untuk melakukan monitoring dan mengukur kinerja seluruh karyawannya terlebih saat karyawan bekerja dari rumah. Untuk mengatasi masalah tersebut, maka diperlukan adanya pemanfaatan teknologi yang telah berkembang pada saat ini dalam mengatasi masalah monitoring dan mengukur kinerja karyawan. Dalam memecahkan masalah tersebut aplikasi *screen monitor duration* berbasis *Support Vector Machine* (SVM) dapat membantu para pemilik perusahaan dalam menjawab masalah. Berdasarkan hasil penelitian yang telah dilakukan, secara garis besar SVM digunakan untuk mengklasifikasikan kinerja karyawan ke dalam kategori produktif atau tidak produktif melalui aplikasi yang dikembangkan. Kemudian metode SVM ini dibandingkan dengan metode *Support Vector Regression* (SVR) untuk mengetahui metode terbaik dalam klasifikasi. Untuk mengetahui hasil akhir ditentukan bahwa klasifikasi terbaik adalah menggunakan metode SVM dibuktikan dengan nilai pengujian *Root Mean Square Error* (RMSE) sebesar 0.4299697 untuk SVM dan sebesar 0.7159644 untuk SVR. Dengan begitu penelitian ini dapat menjawab masalah utama tentang bagaimana cara mengontrol dan monitoring para karyawan, sehingga para pemilik perusahaan dapat mengetahui kinerja karyawan.

Kata kunci: Kinerja Karyawan, Pengembangan Perangkat Lunak, *Root Mean Square Error, Support Vector Machine, Support Vector Regression*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ABSTRACT

EMPLOYEE PERFORMANCE CLASSIFICATION BASED ON SUPPORT VECTOR MACHINE USING SCREEN MONITOR DURATION

By:

Muhammad Aziz Almi

The number of employees in a company makes it difficult for company owners to monitor and measure the performance of all employees, especially when employees work from home. To overcome this problem, it is necessary to use the technology that has developed in overcoming the problem of monitoring and measuring employees performance. Application screen monitor duration based on Support Vector Machine (SVM) can assist company owners in solving the problems. This research resulted in SVM being used to classify employee performance into productive or unproductive categories through the application. Then the SVM method is compared with the Support Vector Regression (SVR) method to find out the best method in classification. In the final result, the best classification was obtained by SVM method as evidenced by the Root Mean Square Error (RMSE) test value of 0.4299697 for SVM and 0.7159644 for SVR. This research can solve the main problem of how to control and monitor employees, so that company owners can find out employees performance.

Keywords: Employee Performance, Software Developer, Root Mean Square Error, Support Vector Machine, Support Vector Regression

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PERSETUJUAN PEMBIMBING ii
PENGESAHAN TIM PENGUJI SKRIPSI iii
PERNYATAAN KEASLIAN iv
UCAPAN TERIMAKASIH v
KATA PENGANTAR vi
ABSTRAK viii
ABSTRACT ix
DAFTAR ISI viii
DAFTAR GAMBAR xii
DAFTAR TABEL xiv
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2 Perumusan Masalah 5
1.3 Batasan Masalah 5
1.4 Tujuan Penelitian 6
1.5 Manfaat Penelitian 6
1.6 Sistematika Penulisan 7
BAB II DASAR TEORI 8
2.1 Penelitian Terdahulu 8
2.2 Teori-teori Dasar 12
2.2.1 Aplikasi 12
2.2.2 Framework 14
2.2.3 Database 15
2.2.4 Model Pengembangan 15
2.2.5 *Web Service* 23
2.2.6 Capture Screen 25
2.2.7 Manajemen Sumber Daya Manusia 25
2.2.8 Machine Learning 26
2.2.9 *Support Vector Machine* 27
2.2.10 Support Vector Regression 31
2.3 Integrasi Keilmuan 33

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III METODOLOGI PENELITIAN 37
3.1 Desain Penelitian 37
3.1.1 Perumusan Masalah 38
3.1.2 Studi Literatur 38
3.1.3 Pengumpulan Data 38
3.1.4 Pengembangan Sistem 39
3.2 Tempat dan Waktu Penelitian 45
3.2.1 Tempat Penelitian 45
3.2.2 Waktu Penelitian 45
BAB IV PEMBAHASAN 46
4.1 Analisis 46
4.1.1 Analisis Kebutuhan Fungsional 46
4.1.2 Analisis Kebutuhan *non Fungsional* 47
4.1.3 Analisis Kebutuhan Perangkat Lunak 47
4.1.4 Analisis Kebutuhan Perangkat Keras 48
4.2 Desain 48
4.2.1 Desain UML 48
4.2.2 Desain *Database* 67
4.3 Implementasi 70
4.3.1 Implementasi SVM 70
4.3.2 Implementasi *User Interface* 74
4.4 Pengujian 79
4.4.1 Pengujian Fungsional 79
4.4.2 Pengujian RMSE 82
4.4.3 Pengujian Performa 84
4.4.4 Perbandingan Metode 85
BAB V KESIMPULAN 87
5.1 Kesimpulan 87
5.2 Saran 88
DAFTAR PUSTAKA 89

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR GAMBAR

Gambar 2.1 *Waterfall Model* 16
Gambar 2.2 Klasifikasi linier data (Christopher D. Manning et al., 2009) 28
Gambar 2.3 Pengukuran lebar margin (Christopher D. Manning et al., 2009) 29
Gambar 2.4 Klasifikasi margin besar (Christopher D. Manning et al., 2009) 30
Gambar 2.5 Intensive Zone 1 32
Gambar 3.1 Alur Penelitian 37
Gambar 3.2 Arsitektur Aplikasi 42
Gambar 4.1 *Usecase Diagram* Penelitian ini 49
Gambar 4.2 *Activity Diagram Login* 50
Gambar 4.3 *Activity Diagram Register* 51
Gambar 4.4 *Activity Diagram Management Project* 52
Gambar 4.5 *Activity Diagram Management Task* 53
Gambar 4.6 *Activity Diagram Monitoring Employee* 54
Gambar 4.7 *Activity Diagram Join Project* 55
Gambar 4.8 *Activity Diagram Tracking* 56
Gambar 4.9 *Activity Diagram Update Task Status* 57
Gambar 4.10 *Sequence Diagram login* 58
Gambar 4.11 *Sequence Diagram Register* 59
Gambar 4.12 *Sequence Diagram Management Project* 60
Gambar 4.13 *Sequence Diagram Management Task* 61
Gambar 4.14 *Sequence Monitoring Employee* 62
Gambar 4.15 *Sequence Diagram Join Project* 62
Gambar 4.16 *Sequence Diagram Tracking* 63
Gambar 4.17 *Sequence Diagram Update Status Task* 64
Gambar 4.18 *Class Diagaram* Penelitian ini 65
Gambar 4.19 Lanjutan *Class Diagaram* Penelitian ini 66
Gambar 4.20 CDM Penelitian ini 67
Gambar 4.21 PDM Penelitian ini 69
Gambar 4.22 *flowchart training process* 72
Gambar 4.23 *flowchart training testing* 73
Gambar 4.24 *Login Web App* 74

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Gambar 4.25 *Register Web App* 75
Gambar 4.26 *Working Board* 75
Gambar 4.27 *management task* 76
Gambar 4.28 *Monitoring page* 76
Gambar 4.29 *Monitoring Employee* 77
Gambar 4.30 *Login Desktop App* 78
Gambar 4.31 *Home Page* 78
Gambar 4.32 *Time Working* 79
Gambar 4.33 *Statistics Error Value* 85
Gambar 4.34 *Statistics RMSE Value* 86

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR TABEL

Tabel 2.1 Simbol-simbol *use case* 17
Tabel 2.2 Simbol-simbol *activity diagram* 18
Tabel 2.3 Simbol-simbol *sequence diagram* 20
Tabel 2.4 Simbol *Class Diagram* 21
Tabel 3.1 Variabel-variabel pada data *screen monitor duration* karyawan 38
Tabel 3.2 Variabel Dataset 40
Tabel 3.3 Fitur yang akan di uji 43
Tabel 3.4 Skenario Pengujian 43
Tabel 3.5 Waktu Penelitian 45
Tabel 4.1 Kebutuhan fungsional 46
Tabel 4.2 Kebutuhan *non Fungsional* 47
Tabel 4.3 Kebutuhan perangkat lunak 47
Tabel 4.4 Kebutuhan perangkat keras 48
Tabel 4.5 Relasi Antar Tabel 68
Tabel 4.6 Dataset setelah tahap *preprocessing data* 71
Table 4.7 Pengujian fitur *login user* 80
Table 4.8 Pengujian fitur *register user* 80
Table 4.9 Pengujian fitur *start working* 81
Table 4.10 Pengujian fitur *result work* 82
Tabel 4.11 Skenario 50: 50 83
Tabel 4.12 Skenario 70: 30 83
Tabel 4.13 Skenario 90: 10 83
Tabel 4.14 Hasil Pengujian Performa 84
Tabel 4.15 Perbandingan Metode 86

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pandemi *Corona Virus Disease 2019* (COVID-19) adalah masalah terbesar dari 200 negara di dunia. Virus ini diidentifikasikan sebagai penyakit pernafasan menular (WHO - World Health Organization, 2020). COVID-19 menyebar ke wilayah negara Indonesia pada tanggal 2 Maret 2020 dengan penyebarannya yang sangat cepat hingga membuat total penderita COVID-19 di Indonesia pada 30 Juli 2021 mencapai sekitar 3.372.374 jiwa (Kemenkes, 2021).

Pada masa pendemi COVID-19 menyebabkan negara Indonesia menetapkan peraturan Pembatasan Sosial Berskala Besar (PSBB) dan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) sehingga menyebabkan para pekerja bertugas dari rumah atau *work from home* (WFH). WFH adalah sistem yang dipilih sebagain perusahaan untuk menjalankan bisnisnya untuk mengurangi penyebaran COVID-19 (Setiati & Azwar, 2020). Namun penerapan WFH tidak semuanya berjalan sesuai harapan dikarenakan tidak semua pekerjaan dapat dilakukan dari rumah. Hal ini menyebabkan banyaknya pemilik perusahaan harus memutar otak dan mencari cara agar pekerjaan-pekerjaan dapat terselesaikan dengan baik dan efektif walaupun dengan segala tantangan yang ada (Akbulaev et al., 2020).

Pelaksanaan WFH ini tidak berdampak pada sektor pekerjaan saja, sektor pendidikan pun ikut terlibat dalam PSBB atau PPKM. Dunia pendidikan kini dilakukan tidak dari sekolah melainkan dari rumah. Oleh karena itu muncul tantangan baru dari sektor pendidikan tentang mengatasi masalah pendidikan yang dilaksanakan dari rumah (García-Peñalvo, 2020). Selain itu, pada sektor pekerjaan pun memiliki tantangan juga terutama bagi pemilik perusahaan tentang bagaimana cara memantau kinerja karyawan dengan adanya jam kerja. Seperti pada pekerjaan sebelum diterapkanya PSBB atau PPKM para karyawan memiliki kontrak kerja mengenai perjanjian kerja bersama dengan kurun waktu yang sudah disepakati dan tidak melanggar hukum (Undang-Undang Cipta Kerja No.11 Tahun 2020). Dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

bekerja dari rumah para pemilik perusahaan ini tidak dapat memantau kinerja karyawan terkait masalah waktu kerja (ILO, 2021).

Kondisi pandemi COVID-19 membuat para pemilik usaha tidak punya pilihan selain memanfaatkan potensi teknologi sepenuhnya untuk melakukan langkah adaptasi (Purwianti & Dila, 2021). Dalam hal ini, maka peran teknologi sangat dibutuhkan dalam segala sektor dikarenakan penggunaan teknologi informasi akan berubah menuju optimalisasi pemanfaatan media digital atau tranfromasi digital. Pemerintah akan mengharuskan masyarakat untuk terlibat dalam penggunaan teknologi untuk terbiasa beradaptasi dengan musim pandemi ini. Kalangan UMKM juga akan dipaksa untuk lebih berkecimpung dalam teknologi dunia digital agar bisnis tetap bisa berjalan walau di masa pandemi (Bakhri & Futiah, 2020).

Terkait permasalahan monitoring karyawan dan kinerja karyawan, banyak aplikasi yang membantu pemilik perusahaan untuk monitoring tugas karyawan seperti *Trello, Slack, Monitask* dan lain-lain dengan perbedaan fitur utama dan metodenya masing-masing tetapi, aplikasi-aplikasi tersebut tidak mampu *screen monitor duration* kerja karyawan. *Screen monitor duration* karyawan ini sangat dibutuhkan oleh pemilik perusahaan dalam menjalankan usahanya dimana pemilik usaha ini akan mengerti bahwa karyawan tersebut sudah bekerja sesuai apa yang dikerjakan dan memenuhi target atau *task* dengan waktu kerja normal yang sudah disepakati seblumnya (Žnidaršic & Bernik, 2021). Setelah dapat *screen monitor duration*, muncul masalah baru apakah selama waktu kerja tersebut karyawan sudah dapat dikategorikan dalam karyawan yang produktif atau sebaliknya. Karyawan produktif dalam hal ini adalah karayawan yang mampu menggunakan waktu kerjanya dengan melakukan hal-hal yang terkait pekerjaannya (Osborne & Hammoud, 2017).

Pada masalah pengembangan aplikasi *screen monitor duration* tentu membutuhkan model pengembangan perangkat lunak seperti analisis, desain dan implementasi. Analisis dan desain ini yang dibutuhkan adalah analisis kebutuhan perangkat lunak, desain *database,* desain *user interface* dan desain sistem (Sommerville, 2011). Dalam pengembangan aplikasi yang serupa *screen monitor*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

duration, banyak metode yang dapat digunakan seperti metode *Expert System* dengan judul penelitian *"Development of Intelligence Process Tracking System for Job Seekers"* (Nanda et al., 2011). Penelitian tersebut menghasilkan sistem pakar yang dirancang bertujuan untuk membuat hubungan yang cerdas antara pengguna atau konsultan dan perusahaan penyedia pekerjaan. *Expert System* dari penelitianya yaitu berusaha mengadopsi pengetahuan para ahli ke sistem, agar sistem mampu mengklasifikasikan pekerjaan sesuai dengan kemampuan (Lucas & van der Gaag, 1991). Dengan begitu untuk pengembangan aplikasi *screen monitor duration* kerja karyawan membutuhkan model pengembangan dan metode yang tepat untuk membantu sistem ini mengambil keputusan dan mengklasifikasikan terkait kinerja karyawan apakah karyawan tersebut masuk kedalam karyawan yang *productive* atau *unproductive*.

Salah satu metode yang digunakan dalam klasifikasi, dapat mengambil dari algoritma pembelajaran mesin seperti *Decision Tree, Boosted Decision Tree, Random Forests, Logistic Regression, Support Vector Machines (SVM),* dan lain-lain. Dari beberapa metode klasifikasi tersebut SVM adalah salah satu metode klasifikasi dengan tingkat akurasi paling tinggi. SVM menghasilkan akurasi 75% dari tiga kategori, sedangkan pada perlakuan yang sama metode-metode lain menghasilkan 53% *decission tree,* 70% *boosted decision tree,* 71% *random forests* dan 73%% *logistic regression* (Beunza et al., 2019). Dengan mempertimbangkan hal tersebut, aplikasi *screen monitor duration* karyawan ini berjalan menggunakan metode SVM dengan pemrosesan data dalam pengambilan keputusan. Tidak hanya itu saja, SVM ini digunakan karena memiliki tujuan untuk menemukan *Hyperplane* terbaik sebagai garis pemisah antara dua buah kelas. Dua buah kelas ini untuk mendefinisikan karyawan tersebut masuk kategori *productive* atau *unproductive.* Metode ini digunakan untuk proses mengklasifikasikan kategori aplikasi yang digunakan lalu menghasilkan sebuah prosentase kinerja karyawan (Kowalczyk, 2017).

Berdasarkan masalah diatas, SVM berjalan menggunakan *supervised learning* dengan fungsi pemetaan agar sistem mengerti ketika ada *variable input (x)* akan menghasilkan *output productive* atau *unproductive.* Setelah itu, aplikasi menerima inputan tersebut dan disimpan ke dalam *database*. Data-data yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersimpan dalam *database* digunakan kembali oleh system untuk menghasilkan *output productive* atau *unproductive* pada sebuah masukan (Abubakar et al., 2016). Sebagai acuan sistem untuk menghasilkan *output,* data-data masukan meliputi pekerjaan karyawan, kategori aplikasi dan aplikasi-aplikasi yang digunakan karyawan saat bekerja. Dengan aplikasi klasifikasi *screen monitor duration* karyawan untuk mengukur kinerja karyawan menggunakan metode SVM akan dapat membantu para pemilik perusahaan dalam menjawab masalah utama tentang bagaimana cara mengontrol dan monitoring para karyawan yang bekerja dari rumah atau pun bekerja dalam perusahaan, sehingga para pemilik perusahaan dapat memonitoring dan mengukur kinerja karyawan (Mustajab et al., 2020). Kinerja karyawan dilihat dari hasil produktivitas kerja karyawan selama menggunakan aplikasi. Nilai kinerja didapatkan melalui perhitungan prosentase dari acuan aplikasi yang digunakan dengan mengklasifikasikan kedalam dua kategori yaitu *productive* dan *unproductive.*

Penelitian ini mengembangkan aplikasi dengan fitur utama *screen monitor duration* karyawan untuk melengkapi beberapa penelitian sebelumnya dengan fitur *capture time, capture application running, automatic screen shoot and productivity result* (Moore, 2000). Aplikasi ini dikembangkan menggunakan framework yang bertujuan untuk mempermudah dan mempercepat pengembangan aplikasi (Veiga et al., 2017). Tidak hanya itu saja, penelitian ini dapat memaparkan tahapan-tahapan dalam pengembangan perangkat lunak dengan model yang digunakan serta penerapan metode SVM dalam klasifikasi.

Klasifikasi ini menggunakan data *training* dalam pelatihan model. Data *training* yang digunakan menggunakan dataset "*Productivity Prediction of Garment Employees Dataset* (https://archive.ics.uci.edu/ml/datasets/)". Dataset ini akan dilakukan *preprocessing* data kemudian memiliki variabel-variabel antara lain jumlah pekerjaan, target produktivitas, *standart minute value, work in progress, over time,* dan *actual productivity*. Pada pelatihan model, SVM memiliki lima tipe yang dapat digunakan yaitu *C-classification, nu-classification, one-classification, eps-regression,* dan *nu-regression.* Pada tipe tersebut dapat disimpulkan bahwa metode SVM ini memiliki dua tipe utama yaitu klasifikasi dan regresi. Penelitian ini juga menggunakan teknik regresi dalam klasifikasi dengan metode *Support*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Vector Regression (SVR). SVR ini adalah salah satu bagian dari SVM yang juga dapat menangani kasus yang sama yaitu klasifikasi (Huang et al., 2008). Oleh karena itu, SVR ini digunakan sebagai metode pembanding untuk penyempurnaan akurasi metode klasifikasi yang digunakan.

Berdasarkan masalah di atas maka dilakukan suatu penelitian yang memfokuskan pada penelitian eksperimen untuk mengukur presentase kinerja karyawan. Dengan begitu, untuk mendapat jawaban dari permasalahan, maka dilakukan penelitian yang berjudul "KLASIFIKASI KINERRJA KARYAWAN BERBASIS SUPPORT VECTOR MACHINE MENGGUNAKAN *SCREEN MONITOR DURATION*". Penelitian ini dapat membantu pemilik perusahaan dengan *monitoring* untuk mengukur kinerja karyawan dan bisa menjadi bahan literasi yang dapat menyempurnakan beberapa penelitian terdahulu.

1.2 Perumusan Masalah

Rumusan masalah ini untuk membantu mengoptimalkan klasifikasi dengan metode SVM pada aplikasi, maka dapat dilakukan:

1. Bagaimana membangun aplikasi *screen monitor duration* karyawan berbasis *desktop*?
2. Bagaimana cara menghasilkan klasifikasi kinerja karyawan menggunakan metode SVM?
3. Bagaimana cara melakukan pengujian pada aplikasi menggunakan metode *Black Box Testing* dan *Root Mean Square Error* (RMSE)?

1.3 Batasan Masalah

Mengingat luasnya ruang lingkup permasalahan yang dihadapi dalam pengembangan aplikasi *screen monitor duration* karyawan, maka penelitian ini membatasi ruang lingkup permasalahan pada:

1. Pengembangan aplikasi ini menggunakan *waterfall model* dengan tahapan analisis, desain, implementasi dan pengujian.
2. Pengembangan aplikasi ini menggunakan *Desktop Application* sebagai *client side* dan *Web Application* sebagai *web service.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3. Teknik klasifikasi menggunakan metode SVM dan metode *Support Vector Regression* (SVR) pada proses pelatihan model.
4. *Dataset* yang digunakan dalam pengolahan data menggunakan dataset "*Productivity Prediction of Garment Employees Dataset* (https://archive.ics.uci.edu/ml/datasets/)".
5. Pengujian *black box testing* berfokus pada fitur-fitur yang berhubungan dengan *role user* meliputi *login, register and tracking time.*
6. Pengujian RMSE menggambil data dari pelatihan model oleh masing-masing metode SVM dan SVR.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini untuk membantu mengatasi permasalahan monitoring kinerja karyawan dengan mengimplementasikan aplikasi dengan tujuan akhir meliputi:

1. Menerapkan langkah-langkah pengembangan aplikasi *screen monitor duration* untuk mengukur kinerja karyawan dengan metode SVM.
2. Menerapkan metode SVM dalam pengolahan data sebagai klasifikasi kinerja karyawan.
3. Mendapatkan nilai presentase fungsi aplikasi yang berjalan sesuai dengan kebutuhan dan presentase perbedaan nilai akurasi dari prediksi.

1.5 Manfaat Penelitian

Manfaat dari penelitian pengembangan aplikasi *screen monitor duration* adalah sebagai berikut:

1. Bagi Akademik

   Penelitian ini dapat digunakan sebagai informasi ilmiah dan studi kasus dalam penelitian selanjutnya yang berkaitan dengan pengembangan aplikasi serupa dan memberikan wawasan bagi pembaca mengenai solusi yang diberikan metode SVM terhadap masalah klasifikasi dan perbandingan metode SVR dalam teknik regresi untuk masalah klasifikasi.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2. Bagi Aplikatif

   Pemilik perusahaan dapat memantau dan mengetahui bagaimana mengukur kinerja karyawannya dan bagi karyawan dapat mengetahui hasil kinerjanya selama bekerja.

1.6 Sistematika Penulisan

Penulisan laporan penelitian ini menggunakan aturan dan sistematika yang dibagi atas beberapa bagian sebagai berikut:

1. Bagian awal penulisan terdiri dari halaman sampul, halaman judul, lembar persetujuan pembimbing, lembar pengesahan oleh pembimbing dan penguji, daftar isi, daftar gambar serta daftar tabel.
2. Bagian isi laporan penelitian terdiri dari:
   a. BAB 1 Pendahuluan, terdiri dari latar belakang, rumusan masalah penelitian yang dilakukan, tujuan penelitian dan manfaat yang diperoleh dari penelitian yang dibuat.
   b. BAB 2 Tinjauan Pustaka, terdiri dari penelitian yang sudah ada dan dilaksanakan serta teori-teori yang terkait dengan penelitian ini seperti konsep pengembangan aplikasi, konsep *maxhibe learning* dan integrasi keilmuan yang sesuai dengan ajaran Islam.
   c. BAB 3 Metodologi Penelitian, terdiri dari alur penelitian dan tempat dan waktu dilaksanakanya penelitian.
   d. BAB 4 Hasil dan Pembahasan, berisi pembahasan terhadap penelitian-penelitian yang dilakukan dalam menyelesaikan rumusan masalah serta menggungkapkan hasil dari penelitian tersebut.
   e. BAB 5 Penutup, terdiri atas kesimpulan dan saran terhadap studi kasus yang dilakukan.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB II

DASAR TEORI

2.1 Penelitian Terdahulu

Sebagai dasar pengetahuan, penelitian ini mengambil beberapa penelitian yang sudah ada dan relevan dengan penelitian sekarang ini dilakukan. Penelitian terdahulu yang diambil berdasarkan pokok bahasan tentang *tracking,* produktivitas karyawan, model pengembangan perangkat lunak, metode SVM, pengujian *fungsional* dan perbandingan metode.

Pada pokok bahasan tentang *tracking,* mengambil penelitian yang dilakukan oleh (Alexander et al., 2017) dengan judul *"Applicant Tracking System (ATS) A Component of HRIS: Benefits and Issues in Its Usage",* menjelaskan tentang sistem pelacakan pelamar pekerjaan. Sistem ini menyediakan cara otomatis bagi perusahaan untuk mengelola seluruh proses perekrutan. *Tracking* dalam penelitian ini yaitu menangkap data-data pelamar kemudian diolah sehingga mendapatkan *output* yang berguna untuk klasifikasi kandidat, menguji pelamar, menjadwalkan wawancara, mengelola proses perekrutan, memeriksa referensi, dan melengkapi dokumen perekrutan baru. Dengan begitu sistem *tracking* ini berperan untuk mengotomatiskan seluruh proses fungsi rekrutmen.

Pada penelitian oleh (Osborne & Hammoud, 2017) dengan judul *"Effective Employee Engagement in The Workplace",* menjelaskan pokok bahasan mengenai produktivitas karyawan. Penelitian itu bertujuan untuk pendalaman strategi yang digunakan pemimpin-pemimpin bisnis komunikasi untuk melibatkan produktifitas karyawan mereka. Temuan penelitian tersebut menunjukkan bahwa ikatan antara pemimpin dan karyawan merupakan elemen penting untuk mewujudkan karyawan yang produktif sehingga dapat meningkatkan profitabilitas organisasi. Pada temuannya lagi karyawan yang tidak produktif biasanya akan membebani perusahaan A.S. $350 miliar per tahun. Karyawan produktif adalah karayawan yang mampu menggunakan waktu kerjanya secara maksimal dengan melakukan hal-hal yang terkait pekerjaannya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada penelitian oleh (Goshu et al., 2017) dengan judul *"Development of Productivity Measurement and Analysis Framework for Manufacturing Companies",* menjelaskan tentang pendekatan alternatif untuk mengukur produktivitas di perusahaan. Penelitian ini bertujuan untuk melakukan penyelidikan mendalam terhadap pengukuran produktivitas yang ada dan praktik analisis kasus perusahaan manufaktur melalui pendekatan kualitatif dan kuantitatif. Dari penelitian ini dapat disimpulkan bahwa pengukuran produktivitas dipengaruhi banyak faktor yang salah satunya antara lain jumlah karyawan dalam satu pekerjaan, jumlah produksi, tingkat kesulitan pekerjaan, dan lama waktu pengerjaan.

Penelitian selanjutnya oleh (Time et al., 2020) dengan judul "*Effective Time Management On Employee Performance Of Northern Nigeria Noodle Company LTD*", menjelaskan dampak manajemen waktu yang efektif terhadap kinerja karyawan dan mengidentifikasi faktor-faktor yang pengaruh manajemen waktu yang efektif terhadap kinerja karyawan. Penelitiannya mengungkap bahwa ada hubungan positif antara kinerja organisasi dan manajemen waktu yang efektif. Berdasarkan temuan, direkomendasikan bahwa peningkatan manajemen waktu yang tepat akan membawa peningkatan yang sesuai dalam kinerja organisasi untuk menciptakan organisasi yang sadar waktu bahwa seseorang harus menjadi lebih efisien waktu. Dengan begitu dapat disimpulkan bahwa manajemen waktu dapat mempengaruhi kinerja karyawan.

Selanjutnya penelitian oleh (Ilag, 2021) dengan judul "*Tools and Technology for Effective Remote Work*", menjelaskan tentang teknologi apa yang efektif digunakan dalam bekerja dirumah. Penelitian ini mengambil contoh dalam bidang komunikasi dan kerjasama dalam bekerja dari rumah menggunakan aplikasi *Microsoft Teams* karena aplikasi tersebut dapat menjembatani kesenjangan komunikasi dan kerjasama secara virtual. Dengan begitu dapat disimpulkan bahwa aplikasi yang efektif adalah aplikasi yang sesuai dengan kebutuhan pekerjaan-pekerjaan yang dilakukan.

Kemudian untuk klasifikasi produktivitas kerja karyawan, penelitian oleh (Mustajab et al., 2020) dengan judul "*Working from Home Phenomenon as an*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Effort to Prevent COVID-19 Attacks and Its Impacts on Work Productivity", menjelaskan dampak bekerja dari rumah pada kinerja karyawan. Hasil dari penelitian ini adalah para karyawan yang tidak produktif melakukan pekerjaan diluar bidang pekerjaanya seperti menggunakan waktu kerjanya tidak untuk bekerja. Begitu sebaliknya, karyawan yang produktif menggunakan waktu kerja dengan melakukan pekerjaanya.

Dari enam penelitian diatas dapat disimpulkan bahwa untuk klasifikasi kinerja karyawan dibutuhkan aplikasi *screen monitor duration* dan data-data terkait produktivitas karyawan yang dilakukan saat waktu bekerja, kemudian dari data tersebut akan diklasifikasi apakah kinerja karyawan tersebut masuk kategori produktif atau tidak produktif.

Kemudian masuk pada pokok pembahasan tentang metode klasifikasi SVM yang diteliti oleh (Abubakar et al., 2016) dengan judul *"A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid",* menjelaskan bagaimana mengklasifikasikan komputasi peta 3D dengan metode SVM. Penelitiannya menggunakan metode SVM untuk secara analitis mengklasifikasikan kemampuan komputasi perangkat seluler yang diperlukan untuk peta 3D yang sesuai digunakan sebagai alat bantu navigasi. Hasil dari penelitian ini menjelaskan terdapat lima puluh ponsel pintar yang berbeda dikategorikan berdasarkan Unit Pemrosesan Grafis (GPU), resolusi layar, memori, dan ukuran. Hasil klasifikasi yang diusulkan menunjukkan akurasi yang tinggi pada percobaan menggunakan SVM.

Setelah itu masuk pada pembahasan tentang pengembangan perangkat lunak pada penelitian oleh (Khoza & Marnewick, 2020) dengan judul *"Waterfall and Agile information system project success rates A South African perspective",* menjelaskan tentang keberhasilan metode pengembangan antara metode *waterfall model* dan *agile.* Hasil dari penelitian itu menerangkan bahwa *waterfall model* sangat banyak dipakai dalam pengembangan perangkat lunak, namun pada *agile* lebih berhasil dari pada metode *waterfall model* akan tetapi metode ini masih ada kekhawatiran yang perlu ditangani mengenai kasus yang dihadapi. Dari pembahasan diatas dapat disimpulkan bahwa pengembangan perangkat lunak

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebelum menentukan metode pengembangan yang akan dipakai, harus mengidentifikasi terlebih dahulu proyek yang akan dibangun seperti apa. Setalah itu mengukur tingkat keberhasilan pada masing-masing metode yang akan digunakan untuk dipakai.

Pada pengembangan perangkat lunak selalu ada tahapan pengujian *prototype*, pengujian ini menguji fungsi-fungsi pada perangkat lunak apakah perangkat lunak tersebut berfungsi sesuai kebutuhan. Membahas tentang pengujian fungsi pada perangkat lunak, penelitian oleh (Akanksha Verma1 et al., 2017) dengan judul *"A Comparative Study of Black Box Testing and White Box Testing",* menjelaskan tentang perbandingan pengujian *black box testing* dan *white box testing.* Pada penelitian tersebut dijelaskan bahwa untuk menguji pada bagian *fungsional* menggunakan metode *black box testing* dan pada bagian *logic* menggunakan *white box testing.* Metode-metode tersebut digunakan untuk pengujian perangkat lunak dengan tujuan akhir untuk meningkatkan kualitas perangkat lunak.

Pengujian dari segi pengukuran perbedaan nilai dari prediksi sebuah model, menggunakan metode *Root Mean Square Error* (RMSE) dari penelitian oleh (Ren et al., 2018) dengan judul *"Study of Dynamometer Cards Identification Based on Root-Mean-Square Error Algorithm",* menjelaskan tentang bagaimana mengidentifikasi *dynamometer* menggunakan dengan pengukuran RMSE. Hasil penelitian tersebut menunjukkan bahwa ketahanan terhadap deformasi grafis algoritma RMSE adalah yang tertinggi di antara semua algoritma. Penerapan algoritma RMSE mengurangi kemungkinan kesalahan identifikasi dan meningkatkan akurasi dan keandalan identifikasi. Dengan begitu dapat disimpulkan bahwa RMSE sangat cocok untuk dijadikan dasar pengujian pengukuran dengan mengukur besaran nilai dari prediksi sebuah model sebagai nilai yang diobservasi.

Sebagai metode pembanding klasifikasi SVM, metode *Support Vector Regression* (SVR) digunakan sebagai klasifikasi juga, seperti pada penelitian oleh (Huang et al., 2008) dengan judul *"Using Support Vector Regression for Classification",* menjelaskan tentang bagaimana menggunakan metode SVR sebagai klasifikasi. Dari penelitian tersebut memaparkan klasifikasi pada kelas

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tarket untuk SVR dibandingkan dengan model M5p-C, SMO dan J48 dengan menerapkan masing-masing metode menggunakan 16 kumpulan data yang diunduh dari UCI pada *platform Weka*. Hasil penelitian menunjukkan bahwa SVR mengungguli M5P-C dan J48, dan menunjukkan kinerja yang komparatif dengan SMO. Fakta ini membuktikan bahwa SVR-C adalah cara baru dan efisien untuk menyelesaikan masalah klasifikasi.

Dari pembahasan diatas, penelitian ini menggabungkan penelitian-penelitian terdahulu yang dapat disimpulkan bahwa *screen monitor duration* berguna untuk menangkap data-data yang lalu kemudian diolah menjadi sebuah *output* yaitu produktivitas karyawan. Lalu untuk mendefinisikan karyawan yang produktif ialah dengan memproses waktu kerja karyawan selama bekerja apakah karyawan tersebut melakukan hal-hal terkait pekerjaanya. Pada tahap selanjutnya metode SVM ini digunakan sebagai klasifikasi karyawan kedalam dua kategori yaitu *productive* dan *unproductive.* Setelah itu aplikasi akan dikembangkan menggunakan konsep SDLC dan pada tahap akhir akan dilakukan pengujian aplikasi dan perbandingan metode.

2.2 Teori-teori Dasar

Teori-teori dasar mencangkup perihal variable-variabel yang digunakan dalam penelitian ini meliputi aplikasi, *framework*, *database,* model pengembangan, *web service, screen monitor duration,* manajemen sumber daya manusia dan *machine learning.*

2.2.1 Aplikasi

Aplikasi adalah data yang sudah diprogram dengan bahasa pemrograman tertentu kemudian disimpan dan diformat secara digital untuk menjalankan fungsi-fungsi tertentu (Filipova, 2018). Aplikasi tidak memiliki bentuk fisik oleh karena itu, aplikasi dapat berjalan ketika ada perangkat fisik atau perangkat keras sebagai media operasi aplikasi tersebut. Aplikasi ini umumnya memiliki tiga jenis yaitu aplikasi berbasis *desktop*, *website* dan *mobile.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. Aplikasi *Desktop*

Aplikasi *desktop* adalah aplikasi yang dijalankan pada laptop atau komputer. Aplikasi *desktop* dikembangkan degan menggunakan bahasa pemrograman seperti *Visual Basic* (VB), *C Sharp* (C#), *Java* dan *Python* (Bustamin, 2021). Aplikasi *desktop* memiliki beberapa kelebihan diantaranya sebagai berikut:

a) *Data Security*

Semua data disimpan dalam sistem *user*. *User* mendapatkan akses penuh atas aplikasi yang telah di *install* pada komputer, dengan begitu data akan mendapatkan perlindungan dari berbagai kerentanan.

b) *Available Controls*

Aplikasi *desktop* memiliki sistem *native control* yang dapat memungkinkan aplikasi mengakses perangkat keras dan komponen *Operating System* (OS) yang dipakai.

c) *Performance*

Aplikasi *desktop* lebih responsive jika dibandingkan dengan aplikasi *website* secara inheren dan berjalan lebih cepat.

Dari penjelasan diatas dapat dijelaskan bahwa aplikasi *desktop* menggunakan lebih sedikit memori sehingga menigkatkan kinerja dan meningkan efisiensi aplikasi. Selain itu aplikasi *desktop* selalu ada pada setiap perangkat keras komputer atau laptop.

2. Aplikasi *Website*

Aplikasi *website* adalah semacam aplikasi yang menggunakkan teknologi penelusuran atau browser untuk membuka dan menjalankan aplikasi yang memerlukan jaringan komputer untuk mengaksesnya (Remick, 2011). Pada pengertian lain aplikasi *website* adalah sebuah program yang disimpan didalam *server* yang memerlukan koneksi internet untuk mengirimnya dan dibuka menggunakkan browser untuk mengaksesnya (Rouse, 2011).

Dari beberapa pengertian oleh para pakar diatas dapat diambil kesimpulan aplikasi web adalah sebuah program yang dapat dijalankan menggunakan *browser* yang memerlukan jaringan internet atau intranet. Progam tersebut juga dibuat dengan kode pemrograman yang mendukung untuk basis *website*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3. Aplikasi *Rstudio*

Aplikasi *Rstudio* adalah aplikasi yang digunakan dalam pengolahan data. Aplikasi ini bersifat terbuka untuk komputasi ilmiah dan teknis (https://www.rstudio.com/). Fungsi dari aplikasi ini untuk membantu pengolahan data dan pelatihan model-model yang kemudian menghasilkan data baru untuk kebutuhan selanjutnya.

2.2.2 Framework

Framework adalah kerangka kerja dengan kemudahan-kemudahan dalam membantu pekerjaan. Kemudahan-kemudahan yang didapatkan akan membantu untuk meningkatkan efektifitas dan efisiensinya yang lebih dalam proses pengembangan suatu perangkat lunak (Veiga et al., 2017). Framework yang digunakan dalam melakukan pengembangan aplikasi ini adalah:

1. *Node JS*

Node JS adalah sebuah perangkat lunak yang berfungsi untuk pengembangan aplikasi *website* dengan konsep yang berbahasa pemrograman *javascript* (https://nodejs.org/). *Node JS* dengan j*avascript* adalah berbeda, perbedaannya adalah jika *javascript* adalah bahasa pemrograman yang difungsikan pada sisi pengguna *browser*, sedangkan *Node JS* digunakan untuk melengkapi *javascript* agar bisa berjalan pada sisi *server* (https://www.javascript.com/).

Javascript adalah bahasa pemrograman yang berbasis *browser*, dimana semua kode ditulis langsung ke dalam *Hypertext Markup Language* (HTML) dari halaman *web*site setelah itu akan diterjemahkan dan dieksekusi untuk mendapatkan respon yang terjadi pada halaman *web*site (Mac Bridem, 2007).

2. *Laravel*

Laravel adalah sebuah *framework* pada aplikasi *website* yang bersifat *open source* dan digunakan untuk membangun sebuah aplikasi website. *Laravel* dibangun berdasarkan pengembangan terhadap bahasa pemrograman *PHP* yang menjadikan lebih cepat dan simple (https://laravel.com/). *Laravel* digunakan sebagian besar *programmer* karena keunggulannya sebagai *framework PHP* karena

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Open-Source, mudah dalam penggunaannya, memiliki *command line/command prompt,* dapat migrasi database dengan sederhana, dan dokumentasinya lengkap.

3. Electron

Electron adalah sistem untuk membuat aplikasi area kerja dengan *javascript* murni dengan memberikan antarmuka pemrograman aplikasi (*Programming Interface*) dari sistem operasi (https://www.electronjs.org/). *Electron* adalah program yang dapat menjalankan *Node JS*, sehingga Electron dapat berjalan di *macOS, Windows*, dan *Linux*.

2.2.3 Database

Database adalah aplikasi yang menyimpan berbagai macam informasi. Setiap kumpulan data memiliki antarmuka pemrograman tertentu untuk membuat, mendapatkan, mengawasi, mencari, dan menduplikasi informasi yang terkandung di dalamnya. Kumpulan data adalah sekumpulan data yang disimpan dalam sebuah PC dengan sengaja sehingga cenderung terlihat menggunakan program PC untuk mendapatkan data dari basis informasi (Solichin, 2010). *Database* digunakan untuk menyimpan data atau informasi yang ada di sekitar yang tergabung dalam PC. Pemanfaatan kumpulan data ini memerlukan pemrograman yang berkapasitas sebagai administrasi kumpulan data atau *Database Management System* (DBMS).

MySQL adalah pemrograman *DBMS multithreaded, multi-klien* dengan sekitar 6 juta perusahaan di seluruh dunia memakai *MySQL*. *MySQL* ini sangat mudah digunakan dan dapat mendukung banyak Bahasa pemrograman (Solichin, 2010).

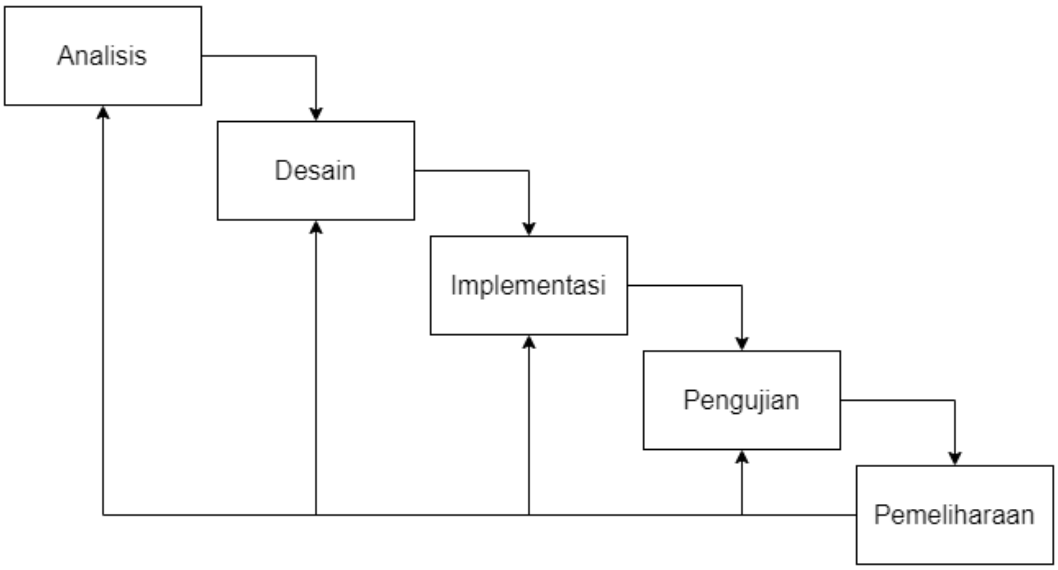
Dari penjelasan diatas dapat disimpulkan bahwa *database* adalah tempat penyimpanan data kemudian *MySQL* yang mengelola data-data yang sudah disimpan pada *database.*

2.2.4 Model Pengembangan

Model pengembangan adalah sebuah rangkaian proses yang berurutan dalam mengembangkan atau membangun perangkat lunak. Salah satu model yang dapat digunakan adalah *waterfall model.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Waterfall model adalah model sistematis dan efisien yang berurutan dalam membangun pemrograman. Model ini mengadopsi strategi yang efisien dan berurutan. Disebut sebagai *waterfall* karena tahapan yang dilalui harus ketat untuk pemenuhan tahapan yang lalu dan berjalan secara berurutan (Sommerville, 2011).



Gambar 2.1 *Waterfall Model* (Sommerville, 2011)

Tahapan-tahapan model pengembangan ini adalah sebagai berikut:

1. Analisis

Analisis adalah kursus data acara sosial tentang kebutuhan pemrograman. Pada tahap ini, prasyarat pemrograman eksplisit dikumpulkan dengan tujuan agar cenderung terlihat kebutuhan pemrograman seperti apa yang dibutuhkan. Kerangka analisis kebutuhan dibagi menjadi dua:

- Kebutuhan Fungsional

Kebutuhan fungsional adalah penegasan administrasi yang harus diberikan kepada kerangka kerja untuk menggunakan kebutuhan untuk menanggapi sumber informasi tertentu dan dalam keadaan tertentu (Kosasi dan Kuway, 2012). Kebutuhan fungsioanl ini harus dipenuhi dalam pengembangan aplikasi.

- Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan opsional yang dapat digunakan ataupun tidak digunakan dengan sistem harus memenuhi kebutuhan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

fungsional terlebih dahulu (Kosasi & Kuway, 2012). Kebutuhan non-fungsional ini mencakup kebutuhan perangkat keras dan perangkat lunak yang digunakan.

2. Desain

Penyelidikan prasyarat pemrograman diubah menjadi struktur rencana sehingga cenderung dieksekusi menjadi program pada tahap selanjutnya. Hal-hal yang dilakukan dalam tahap desain adalah rencana kerangka kerja yang memanfaatkan *Unified Modelling Language* (UML).

UML adalah bahasa visual sebagai outline dan teks untuk menentukan, menggambarkan, merakit, dan mengarsipkan pemrograman (Rosa dan Salahuddin, 2011). Dalam arti lain, UML adalah dokumentasi grafis yang digunakan dalam menggambarkan dan merencanakan kerangka kerja yang dibuat menggunakan strategi pemrograman yang diatur objek (Martin Fowler, 2005). Sebagian dari bagan UML ini adalah:

a. Use Case Diagram

Grafik *use case* menggambarkan kegunaan normal dari suatu kerangka kerja, yang dikendalikan oleh apa yang dilakukan kerangka tersebut. Kasus pemanfaatan membahas koneksi antara penghibur dan kerangka kerja. Perlunya penamaan kasus pemanfaatan adalah nama yang sangat mudah digunakan.

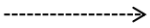
Aktor untuk kasus penggunaan ditampilkan terhubung ke kasus penggunaan di mana mereka terlibat (Bridgeland & Zahavi, 2009). Aktor dalam *use case* sering digunakan sebagai pengguna aplikasi, dengan begitu aktor ini akan berinteraksi dengan proses-proses *case* yang terlibat dalam system tersebut.

Desain *use case* digambarkan dengan susunan simbol-simbol yang dibuat secara konsep sistem berjalan. Simbol-simbol *use cae* dapat dilihat pada Table 2.1

Tabel 2.1 Simbol-simbol *use case* (Wesley, 2015)

Simbol	Deskripsi
Case	Rangkaian proses atau aktivitas yang dilakukan oleh aktor

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Aktor	Object yang berinteraksi dengan *case* kegiatan yang dilakukan
Asosiasi	Menunjukkan korespondensi atau hubungan antara aktor dan *case* atau antara kasus aktor
Extend	Hubungan *case* di mana *case* pemanfaatan tambahan dapat tetap soliter bahkan tanpa *case* penggunaan tambahan.
Include 	Koneksi antara *case* penggunaan tambahan ke situasi pemanfaatan di mana kasus pemanfaatan tambahan memerlukan kasus pemanfaatan ini untuk memainkan kapasitasnya

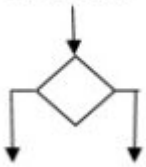
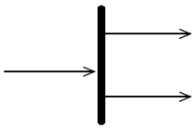
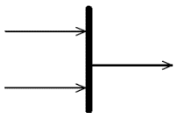
b. Activity Diagram

Activity diagram adalah penggambaran perkembangan pekerjaan atau latihan dari suatu kerangka kerja. *Activity diagram* juga menggambarkan bagaimana alur kerangka kerja dimulai, keputusan (pilihan) yang mungkin terjadi dan lebih jauh lagi bagaimana alur kerangka kerja ditutup (Sugiarti, 2013).

Tabel 2.2 Simbol-simbol *activity diagram* (Wesley, 2015)

Simbol	Deskripsi
Initial State	Awal dimulainya kegiatan pada *activity diagram*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

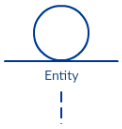
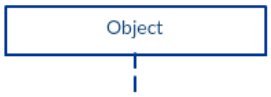
Final State	Bagian akhir kegiatan pada *activity diagram*
Activity	Aktivitas atau proses yang dilakukan
Decision 	Aktivitas pilihan kondisi yang mugkin terjadi sehingga proses dapat berjalan dengan lebih dari satu
	Memecah *behavior* menjadi aktivitas paralel
	Menggabungkan Kembali aktivitas paralel

c. Sequence Diagram

Sequence diagram berpusat di sekitar perilaku di dalam kerangka kerja, menunjukkan bagaimana proses berinteraksi dengan prose yang berbeda. *Sequence diagram* digunakan untuk menggambarkan asosiasi proses yang terjadi dalam kasus penggunaan. Satu kasus hanya membutuhkan satu *sequence diagram*, jika ada beberapa situasi dalam kasus pemanfaatan, mereka dapat diuraikan sebagai bagian dalam bagan suksesi (Cruz-Lemus, 2011). Dalam pengujian pemrograman terkemuka, *sequence diagram* juga memainkan peran penting, terutama sebagai alasan untuk melakukan pengujian (Swain, 2010).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 2.3 Simbol-simbol *sequence diagram* (Wesley, 2015)

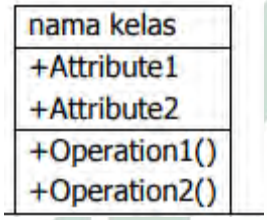
Simbol	Deskripsi
	Penggambaran aktor yang ada dan berinteraksi dengan proses-proses dalam sistem
	Penggambaran hubungan proses yang akan dilakukan oleh sebuah objek
	Penggambaran aktivitas dalam pengisian formulir
	Menggambarkan hubungan antara *boundary* dengan tabel
	Pengiriman pesan
	Menggambarkan object yang akan berinteraksi

d. Class diagram

Class diagram adalah penggambaran struktural dalam objek-objek yang digunakan oleh sistem. *Class diagram* juga memperlihatkan bagaimana hubungan antar objek satu dengan objek yang lainya (Sugiarti, 2013). Simbol-simbol yang digunakan dalam *class diagram* seperti pada tabel 2.4

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 2.4 Simbol *Class Diagram* (Wesley, 2015)

Simbol	Deskripsi
Package	Menggambarkan sebuah wadah dari satu atau lebih objek didalamnya
Kelas 	Menggambarkan isi dari setiap kelas yang meliputi nama kelas, atribut dan juga fungsi-fungsi yang digunkana
Asosiasi	Menggambarkan hubungan anatara kelas satu dengan kelas yang lainya

Pada tahapan desain tidak hanya melakukan desain pada sistem saja, selanjutnya akan diteruskan dengan desain *database.* Desain *database* digunakan untuk mempermudah mengatur lemari penyimpanan dengan tepat supaya pengolahan data akan menjadi efektif. Pada tahapan ini, *database* didesain melalui dua tingkatan yaitu:

- *Conceptual Data Model* (CDM)

CDM adalah desain yang menggambarkan model data antar tabel secara konseptual. Pada CDM model data masih berupa nama tabel, atribut dan tipe data, *primary key,* dan relasi hubungan antar tabel. Desain pada CDM ini masih tidak terdapat *foreign key.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

- *Physical Data Model* (PDM)

PDM adalah desain yang menggambarkan model data tingkat terakhir kemudian akan dapat di ubah menjadi SQ. Perbedaan antara desan CDM dan desain PDM yaitu pada CDM tidak mimiliki *foreign key*, sedangkan pada PDM terdapat *foreign key* hubungan antara tabel satu dengan yang lain.

3. Pengujian

Pengujian perangkat lunak (*software testing*) berperan penting dalam sistem informasi untuk melakukan suatu analisis mengenai permasalahan dalam perangkat lunak. Pengujian ini digunakan untuk menemukan kesalahan atau *error* yang muncul dari sistem perangkat lunak yang sedang digunakan. Adanya pengujian seorang *programmer* atau pengembang aplikasi dapat mengetahui apa yang harus diperbaiki supaya aplikasi dapat berjalan sesuai keinginan.

Dari penjelasan yang sudah dipaparkan di atas, pengujian perangkat lunak adalah elemen atau komponen kritis yang digunakan untuk menentukan kualitas suatu perangkat lunak (Jaya, 2018). Pengujian tersebut meliputi *coding, design,* dan spesifikasi dalam perangkat lunak. Dalam artian lain pengujian adalah salah satu tugas *software verification and validation* yang merupakan bagian dari s*oftware quality assurance* (Rouf, 2012). Berikut beberapa metode yang digunakan dalam pengujian perangkat lunak:

a. *Black Box Testing*

Metode *Black Box Testing* adalah suatu teknik pengujian yang berfokus menguji spesifikasi fungsional yang ada pada perangkat lunak dengan mengumpulkan kondisi masukan dan melakukan suatu pengujian pada spesifikasi fungsional program (Fauzi et al., 2020). Teknik ini mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan *fungsional* untuk suatu program kemudian diuji sesuai skenario pengujian.

b. *Root Mean Square Error* (RMSE)

RMSE adalah skor kuadrat kesalahan atau perbedaan antara harga asli dengan nilai yang diharapkan, kemudian, pada saat itu, membagi angka itu dengan kesempatan dan ukuran dan kemudian mengambil akarnya. RMSE sendiri

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

merupakan strategi elektif untuk menilai strategi ekspektasi yang digunakan untuk mengkuantifikasi ketepatan ukuran konsekuensi suatu model (H. Lau et al., 2015). Nilai RMSE berikutnya adalah nilai normal kuadrat dari kuantitas kesalahan dalam model peramalan. RMSE adalah strategi yang tidak sulit untuk diterapkan dan telah digunakan secara umum dalam berbagai pengujian yang diidentifikasi dengan harapan atau pengukuran (Ren et al., 2018) dengan persamaan sebagai berikut:

$$\text{RMSE}=\sqrt{\frac{\sum(\text{Aktual-Predikat})^2}{n}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.1)$$

Dimana:

Aktual adalah nilai hasil data aktual

Predikat adalah nilai hasil prediksi

N adalah jumlah data

∑ adalah *summation* (jumlah keseluruhan nilai)

Nilai RMSE terendah artinya variasi nilai yang dihasilkan oleh suatu model prakiraan mendekati variasi nilai obeservasinya. RMSE menghitung seberapa berbedanya seperangkat nilai prediksi dengan aslinya. Semakin kecil nilai RMSE, semakin dekat nilai yang diprediksi dan diamati (Gunawan et al., 2016).

RMSE digunakan dalam investigasi serupa dari informasi lapangan dengan informasi yang ditangani untuk memutuskan tingkat kesalahan yang terjadi. RMSE adalah nilai normal dari jumlah kuadrat kesalahan atau jumlah kuadrat dari kualitas yang diantisipasi dan diperhatikan. RMSE juga dapat mengungkapkan ukuran kesalahan yang dihasilkan oleh model pengukur.

2.2.5 *Web Service*

Web Service adalah teknik komunikasi antara sistem komputer yang berbeda dengan artian memiliki *environment* yang berbeda. Dengan adanya *web service* komunikasi antara system yang berbeda dapat dilakukan (Ethan Cerami, 2002). *Web service* secara umum disusun dengan tiga komponen sebagai berikut:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. *Simple Object Acces Protocol (SOAP),* sebagai protocol yang berfungsi untuk pertukaran informasi melalui jaringan tersebar.
2. *Web Service Definition Languge (WSDL),* dokumen standar yang digunakan *web service* dituliskan menggunakan format *extensible markup language* (XML).
3. *Universal Deskription, Discovery, and Integration (UDDI),* pengaksesan XML yang ada dalam jaringan yang berisi *service* dan bersifat bebas.

Repsentational state transfer (*RESTful*) adalah salah satu jenis *web service* yang sering digunakan karena layanan yang diberikan sangat sederhana dan ringan. *Web service* API yang menggunakan REST disebut RESTful API (Gaston C. Hillar, 2018). RESTful API tidak memiliki standar notasi yang resmi, oleh karena itu RESTful API dikatakan sebagai arsitektur.

Application Programming Interface (API) adalah sebuah perangkat lunak yang digunakan oleh *developer* dalam hal mengintegrasikan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain. Dua aplikasi yang terhubung akan saling melakukan pertukaran data dengan format JSON.

JavaScript Object Notation (JSON) adalah format notasi untuk interaksi data. JSON digunakan untuk menjadi pertukaran data yang mudah dipahami oleh seorang *developer* untuk mengelola data JSON tersebut (https://www.json.org/). JSON didukung langsung di dalam *JavaScript* dengan contoh data sebagai berikut:

```
{
        "application_id": 1,
        "project_id": 1,
        "total_time": "08:20:10"
        "status": true
}
```


digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.2.6 Capture Screen

Capture screen adalah proses penangkapan suatu objek yang dapat diketahui ketika sudah dilakukan atau sedang dilakukan saat ini. *Capture screen* dapat berupa waktu, posisi, aktivitas dan lain-lain.

1. Capture time

Capture time adalah suatu proses penangkapan yang melibatkan waktu yang digunakan untuk dicatat dan dihitung berapa banyak waktu yang digunakan dalam melakukan aktivitas tersebut.

2. Capture application running

Capture application running adalah suatu proses penangkapan sebuah aplikasi yang sedang berjalan atau sedang dibuka. Proses ini menangkap nama aplikasi apa saja yang sedang berjalan.

3. Screen shoot

Screen shoot adalah suatu proses menangkap layar atau tampilan pada layar atau halaman yang dilihat pengguna. Kegiatan ini akan dijalankan untuk mengambil tampilan layar karyawan.

2.2.7 Manajemen Sumber Daya Manusia

Pada sebuah perusahaan, Sumber Daya Manusia (SDM) dinilai sebagai asset yang sangat penting melebihi sumber daya lainya. Manajemen SDM adalah suatu kegiatan berkaitan dengan bagaimana merencanakan, menempatkan, mengkoordinasi dan mengontrol SDM yang bekerja dalam organisasi (Michael, 2017).

Dari pemahaman mengenai manajemen SDM dapat disimpulkan bahwa SDM memiliki pengaruh dalam susksesnya sebuah perusahaan. Oleh karena itu, untuk meningkatkan kinerja perusahaan maka dukungan dari manajemen SDM sangatlah penting. Setelah itu, kinerja karyawan menjadi focus utama dalam mendukung operasional manajemen SDM.

Kinerja karyawan memegang peranan yang sangat penting dalam tercapinya tujuan dari suatu perusahaan. Perusahaan yang memiliki produktivitas karyawan yang tinggi dapat dengan mudah dalam mencapai tujuan yang diinginkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sedangkan suatu perusahaan tidak dapat mencapai tujuan yang diinginkan apabila produktivitas karyawan yang dimiliki perusahaan tersebut rendah. Dengan begitu untuk meningkatkan produktivitas karyawan, perusahaan perlu memberikan pelatihan untuk para kayawannya.

Produktivitas adalah pendekatan interdisipliner dalam menentukan mutu kehidupan yang efektif, pembuatan rencana, aplikasi penggunaan cara yang produktifitas untuk menggunakan sumber-sumber secra efektif dan efisien dan tetap menjaga kualitas yang tinggi (Misbahun Deden Muayyad & Irma Ade Oktafia Gawi, 2016).

Dalam meningkatkan produktivitas maka perlu memperhatikan beberapa hal yang dapat menjadi potensi yang lebih tinggi diantaranya adalah pengurangan jam kerja yang tidak efektif, lamanya dalam bekerja, pelatihan, dan motivasi. Pada umumnya produktivitas terbuang percuma disebabkan oleh beberapa hal diantaranya:

1. Struktur waktu kerja

Kemampuan manajemen dan pengawasan dala mengurangi sebab-sebab utama dari kerugian waktu serta dalam perencanaan teknik-teknik dalam meningkatkan produktivitas bagi kepentingan individu

2. Penggunaan waktu kerja dalam meningkatkan efektivitas

Kemampuan masing-masing individu dan sikap saat bekerja dapat dijadikan sebagai upaya dalam meningkatkan produktivitas kerja. Beberapa sikap yang harus diketahui diantaranya yaitu pada aspek jenjang pendidikan, jenis teknologi tang dapat digunkan, kondisi kerja, kesehatan fisik maupun mental.

Dari penjelasan diatas dapat disimpulkan bahwa aplikasi *capture screen* monitor adalah pelacakan terhadap kemampuan *capture* apa yang dikerjakan karyawan dan penggunaan waktu kerja dalam melakukan aktifitas pekerjaan.

2.2.8 Machine Learning

Machine learning (ML) adalah mesin cerdas yang dapat belajar sendiri tanpa ada arahan dari penggunanya dan dikembangan dengan teori pembelajaran komputasi dalam kecerdasan buatan (Alex Smola & S.V.N. Vishwanathan, 2008).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ML berjalan dengan belajar menganalisa data berdasarkan *dataset* atau data-data masukan yang digunakan pada awal dan juga dapat pada saat ML digunakan. ML memiliki dua teknik dasar dalam hal belajar yaitu:

1. Teknik *supervised learning*

Teknik *supervised learning* adalah metode yang diterapkan pada pembelajaran mesin dengan mengakui data yang sekarang ada dalam informasi dengan menunjuk tanda tertentu. Metode ini sekaligus memberikan tujuan terhadap hasil yang dilakukan dengan melihat pertemuan-pertemuan pembelajaran yang dilakukan (Sathya & Abraham, 2013).

2. Teknik *Unsupervised Learning*

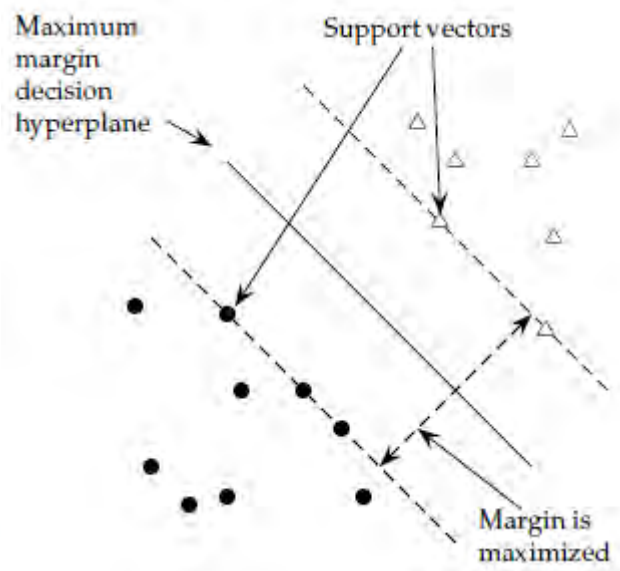
Teknik *Unsupervised Learning* adalah metode yang diterapkan pada pembelajaran mesin yang digunakan pada informasi yang tidak memiliki data yang dapat diterapkan secara langsung. Metode ini digunakan untuk membantu menemukan konstruksi atau contoh yang disimpan dalam informasi yang tidak bervariabel (Alex Smola & S.V.N. Vishwanathan, 2008).

Dari teknik pembelajaran ML juga memiliki banyak metode yang dapat diterapkan pada kasus-kasus tertentu, sebagai contoh pada kasus klasifikasi dapat menggunakan metode *Support Vector Machine* (SVM) dan *Support Vector Regression* (SVR).

2.2.9 *Support Vector Machine*

SVM adalah sejenis pengklasifikasi margin besar. SVM ini adalah metode pembelajaran mesin berbasis ruang vektor di mana tujuannya adalah untuk menemukan batas keputusan antara dua kelas yang secara maksimal jauh dari titik mana pun dalam data pelatihan (Christopher D. Manning et al., 2009). Klasifikasi penggunaan SVM untuk kasus set data dua kelas yang dapat dipisahkan oleh pengklasifikasi linier seperti pada Gambar 2.2

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.2 Klasifikasi linier data dua kelas (Christopher D. Manning et al., 2009)

SVM secara khusus mendefinisikan kriteria untuk mencari permukaan keputusan yang secara maksimal jauh dari titik data mana pun. Kemudian jarak dari permukaan keputusan ke titik data terdekat menentukan *margin* dari pengklasifikasi. Pada Gambar 2.2 menunjukkan bahwa vektor margin dan dukungan untuk masalah sampel titik data lain tidak berperan dalam menentukan permukaan keputusan yang dipilih (Christopher D. Manning et al., 2009).

Memaksimalkan margin mewakili keputusan klasifikasi yang sangat tidak pasti hampir 50% kemungkinan klasifikasi memutuskan salah satu cara. Pengklasifikasi dengan margin besar tidak membuat keputusan klasifikasi dengan kepastian rendah. Dengan begitu margin keamanan klasifikasi sedikit kesalahan dalam pengukuran atau sedikit variasi data tidak akan menyebabkan kesalahan klasifikasi.

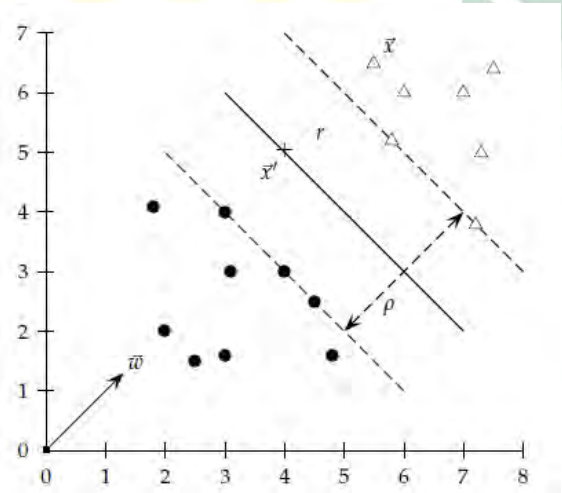
Mempersamakan SVM pada aljabar dengan *hyperplane* keputusan dapat didefinisikan dengan variabel b dan vektor normal *hyperplane* keputusan yang tegak lurus terhadap *hyperplane* tersebut. Dengan begitu untuk memilih di antara semua hyperplane yang tegak lurus terhadap vektor normal dengan menentukan variable b karena *hyperplane* tegak lurus terhadap vektor normal, semua titik $\bar{x}$ pada hyperplane memenuhi $\vec{w}^T\vec{x} = -b$. Misal titik data pelatihan $D = \{(\vec{x_i}, y_i)\}$, di mana setiap anggota adalah pasangan titik $\vec{x_i}$ dan label kelas y_i yang sesuai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengannya. Dengan begitu untuk SVM, dua kelas data selalu diberi nama $+1$ dan -1 (bukan 1 dan 0), dan suku variabel selalu direpresentasikan secara eksplisit sebagai b (Christopher D. Manning et al., 2009). Dari situ dapat dituliskan persamaan untuk definisi margin fungsional klasifikasi linier sebagai berikut:

$$f(x)=\text{sign}(w^T x+b) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$

Nilai -1 menunjukkan satu kelas, dan nilai $+1$ menunjukkan kelas lainnya. Setelah itu untuk kumpulan data dan *hyperplane* keputusan, dapat mendefinisikan *margin fungsional* dari contoh ke i^{th} sebagai contoh $\vec{x_i}$ sehubungan dengan *hyperplane* $(\vec{w}, b)$ sebagai kuantitas $y_i(w^T x + b)$ (Christopher D. Manning et al., 2009). Margin fungsional dari kumpulan data sehubungan dengan permukaan keputusan kemudian dua kali margin fungsional dari salah satu titik dalam kumpulan data dengan margin fungsional minimal yaitu faktor 2 berasal dari pengukuran di seluruh lebar margin, seperti pada Gambar 2.2.



Gambar 2.2.3 Pengukuran lebar margin (Christopher D. Manning et al., 2009)

Pada Gambar 2.3 dapat disimpulkan bahwa variable r menyatakan jarak. Jarak terpendek antara titik dan *hyperplane* tegak lurus terhadap bidang, dan karenanya, sejajar dengan $\vec{w}$. Vektor satuan dalam arah ini adalah $\vec{w}/|\vec{w}|$. Garis putus-putus pada diagram adalah terjemahan dari vektor $r\vec{w}/|\vec{w}|$. Setelah itu beri label titik pada *hyperplane* yang paling dekat dengan $\vec{x}$ sebagai $\vec{x}'$. Kemudian dipersamaankan sebagai berikut:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$\vec{x}' = \vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|} \quad \text{..... (2.3)}$$

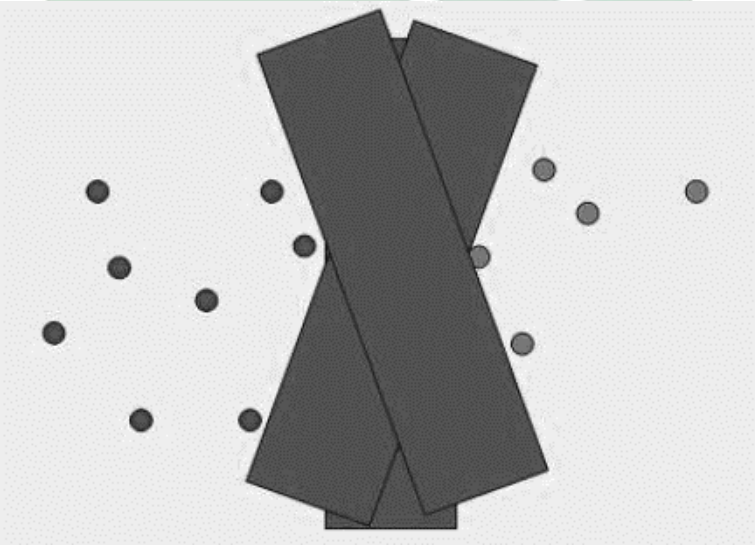
dengan mengalikan dengan y hanya mengubah tanda untuk dua kasus $\vec{x}$ berada di kedua sisi permukaan keputusan. Selain itu, $\vec{x}'$ terletak pada batas keputusan dan memenuhi $\vec{\omega}^T\vec{x}' + b = 0$. Oleh karena itu:

$$\vec{\omega}^T\left(\vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|}\right) + b = 0 \quad \text{..... (2.4)}$$

pemecahan untuk r memberikan:

$$r = y\frac{\vec{\omega}^T\vec{x} + b}{|\vec{\omega}|} \quad \text{..... (2.5)}$$

seperti halnya yang tadi, titik-titik yang paling dekat dengan hyperplane pemisah adalah vektor pendukung (Christopher D. Manning et al., 2009). Margin geometris dari pengklasifikasi adalah lebar maksimum pita yang dapat ditarik untuk memisahkan vektor pendukung dari dua kelas, artinya ini adalah dua kali nilai minimum di atas titik data untuk r yang diberikan dalam Persamaan 2.6 atau secara ekuivalen, lebar maksimal dari salah satu pemisah yang ditunjukkan pada Gambar 2.4.



Gambar 2.4 Klasifikasi margin besar (Christopher D. Manning et al., 2009)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

pada Gambar 2.4 margin geometris jelas invarian terhadap penskalaan parameter jika mengganti $\vec{w}$ dengan $5\vec{w}$. dan b dengan $5b$, maka margin geometrinya sama, karena secara inheren dinormalisasi dengan panjang $\vec{w}$.

Dari penjelasan diatas dapat disimpulkan bahwa SVM dapat menerapkan batasan penskalaan apa pun yang yang menjadi kebutuhan dari masalah yang dihadapai dengan pengaruh pada $\vec{w}$ tanpa mempengaruhi margin geometrik.

2.2.10 Support Vector Regression

Perhitungan *Support Vector Regression* (SVR) adalah hipotesis yang disesuaikan dari hipotesis ML yang telah digunakan untuk menangani masalah penataan *Support Vector Machine* (SVM). SVR ini adalah eksekusi perhitungan SVM. Strategi SVM adalah penggunaan hipotesis ML untuk karakterisasi kasus yang menghasilkan kualitas bilangan, sedangkan perhitungan SVR adalah untuk pemanfaatan kasus regresi yang menghasilkan nilai sebagai bilangan asli (Furi, Jordi, & Saepudin, 2015).

Tujuan SVR adalah untuk menemukan kapasitas sebagai *hyperplane* sebagai pekerjaan regresi yang cocok untuk semua informasi dengan kesalahan dan membuatnya tipis seperti yang diharapkan (Scholkopt and Smola, 2002). Motivasi di balik SVR ini adalah untuk merencanakan vektor informasi ke dalam aspek yang lebih tinggi (Abe, 2005). Misalkan sebuah fungsi berikut adalah garis regresi sebagai optimal *hyperplane*:

$$f(x,a)=w.\varphi(x)+b \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

Dimana:

a = parameter *machine learning*
φ = peta dari ruang asli x ke tinggi a
b = ambang batas

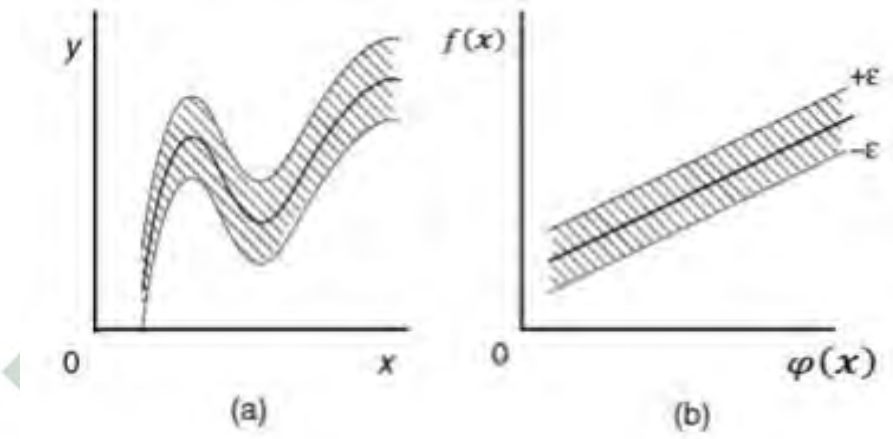
interpretasi y hanya mngambil nilai -1 dan +1. Masalah *learning* disebut sebagai *Support Vector Classification* (SVC). Jika tidak, jika domain y berisi kontinu nilai sebenarnya, yaitu *Support Vector Regression* (SVR). Dengan memperkenalkan fungsi kernel:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$K(x,y)=\varphi(x) \cdot \varphi(y) \quad \text{..... (2.7)}$$

Pada regresi terdapat residual *(r)* kemudian residual ini didefinisikan dengan mengurangkan output scalar *(y)* terhadap estimasi *f(x)* dirumusakan sebagai $r = y - f(x)$ dengan:

$$E(r) = \begin{cases} 0 & \text{untuk } |r| \leq \in \\ |r| - \varepsilon & \text{untuk yang lain} \end{cases} \quad \text{..... (2.8)}$$



Gambar 2.5 Intensive Zone 1 (Scholkopt and Smola, 2002).

(a) *Original input space* dan (b) *feature space*

$D(x,y) = \pm\varepsilon$ adalah jarak terjauh *support vector* dan *hyperplane*, kemudian disebut margin. Memaksialkan margin akan meningkatkan probabilitas data ke dalam radius $\pm\varepsilon$. Jarak dari *hyperplane* D(*x,y*) = 0 ke data (*x,y*) adalah $|D\ (x,y)|/\|w°\|$, dimana :

$$\boldsymbol{w}° = (1\text{-}w^T)^T \quad \text{..... (2.10)}$$

Diasumsikan bahwa jarak maksimum data terhadap *hyperplane* adalah δ, maka estimasi yang ideal terpenuhi dengan:

$$\frac{|D\ (x,y)|}{\|w°\|} \leq \delta \quad \text{..... (2.11)}$$

$$|D\ (x,y)| \leq \delta\|w°\|$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$\delta \|w^{\circ}\| = \varepsilon$$

Oleh karena itu untuk memaksimalkan margin δ, diperlukan $\|w^{\circ}\|$ yang minimum. Optimasi penyelesaian masalah dengan bentuk *Quadratic Programming* dapat dirumuskan:

$$\text{Min } \frac{1}{2}\|w\|^2 \quad \ldots\ldots\ldots\ldots (2.12)$$

Dengan syarat

$$y_i - w^T\varphi(x_1) - b \leq \varepsilon \; untuk \; i = 1, \ldots\ldots, l$$
$$w^T\varphi(x_1) - y_1 + b \leq \varepsilon \; untuk \; i = 1, \ldots\ldots, l$$

Perhitungan SVR dapat menciptakan nilai penentu yang besar karena SVR dapat mengatasi masalah *overfitting* (Furi, Jordi, dan Saepudin, 2015). *Overfitting* adalah pelaksanaan informasi selama tahap persiapan atau persiapan yang menghasilkan presisi harapan yang praktis luar biasa (Yasin, Prahutama, dan Utami, 2014). Tujuan dari perhitungan SVR adalah untuk mencari garis isolasi yang pasti dapat diketahui sebagai *Hyperplane* terbaik. *Hyperplane* terbaik dapat ditemukan dengan memperkirakan margin dengan *hyperplane* tersebut. Tepi sebenarnya adalah pemisahan dari hyperplane ke informasi terdekat. Informasi yang paling dekat dengan *edge* dikenal sebagai *support vector* (Furi, Jordi, & Saepudin, 2015).

2.3 Integrasi Keilmuan

Seperti yang diterangkan oleh ustadz Sulaiman dari Mojokerto selaku penceramah. Beliau menjelaskan bahwa aplikasi *capture screen* monitor ini memanfaat waktu dan aplikasi yang digunakan karyawan dalam melakukan pekerjaanya hingga mendapatkan hasil kinerja karyawan tersebut. Membahas tentang perilaku karyawan saat bekerja tidak luput dengan perilaku tidak jujur. Salah satu contoh perilaku tidak jujur dalam sektor pekerjaan adalah korupsi.

Korupsi seringkali dihubungkan dengan kerugian berupa materi atau uang. Namun dalam pernggunaan ragam bahasa korupsi tidak hanya berupa materi. Korupsi juga dapat berupa waktu seperti waktu kerja yang digunakan untuk urusan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

pribadi. Korupsi waktu terjadi ketika seseorang lalai dalam menjalankan amanah waktu yang telah disepakati.

Pada penelitian oleh (Indra Mayesti, 2016) menjelaskan bahwa apabila ada seorang Pegawai Negeri Sipil (PNS) yang korupsi waktu selama 1 jam saja, Negara dirugikan sebasar Rp. 14 juta rupiah. Berdasarkan kalkulasi tersebut maka dapat dikatakan bahwa korupsi waktu yang dilakukan oleh seorang pekerja di dalamnya terdapat korupsi uang. Apabila seorang karyawan melakukan korupsi waktu maka sama halnya dengan memakan harta yang bukan hak nya. Sebagai seorang pekerja terutama seorang muslim tidak boleh hanya menuntut gaji namun dalam menjalankan amanah yang telah disepakati tidak boleh lalai. Hal ini akan menyebabkan seorang pekerja tersebut memakan harta yang tidak halal atau bathil dan berakibat pada harta yang tidak berkah. Firman Allah SWT dalam surat Al Baqarah ayat 188 yang berbunyi,

وَلَا تَأْكُلُوْٓا اَمْوَالَكُمْ بَيْنَكُمْ بِالْبَاطِلِ وَتُدْلُوْا بِهَآ اِلَى الْحُكَّامِ لِتَأْكُلُوْا فَرِيْقًا مِّنْ اَمْوَالِ النَّاسِ بِالْاِثْمِ وَاَنْتُمْ تَعْلَمُوْنَ

Artinya:

"Dan janganlah kamu makan harta di antara kamu dengan jalan yang batil, dan (janganlah) kamu menyuap dengan harta itu kepada para hakim, dengan maksud agar kamu dapat memakan sebagian harta orang lain itu dengan jalan dosa, padahal kamu mengetahui".

Ayat diatas menjelaskan bahwa Allah SWT melarang umat manusia mengambil sesuatu yang bukan haknya. Sebab itu, mengambil sesuatu yang bukan haknya hukumnya haram, baik dengan mencuri, merampas, menipu, berlaku curang ataupun pemalsuan. Hal-hal tersebut masuk dalam golongan korupsi. Korupsi adalah tindakan dilarang oleh Allah SWT yang sudah dijelaskan pada ayat diatas. Korupsi bukan hanya menyangkut uang saja, melainkan waktu ada kaitanya dengan masalah korupsi.

Waktu atau masa merupakan keseluruhan dari rangkaian saat terjadinya proses, perbuatan, atau kondisi yang sedang berlangsung. Waktu merupakan salah satu nikmat tertinggi yang diberikan oleh Allah SWT kepada manusia. Dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

begitu itu harusnya manusia memanfaatkan waktu yang diberikan oleh Allah SWT dengan sebaik mungkin untuk menjalankan kewajibannya sebaik mungkin. Allah SWT berkali-kali bersumpah menggunakan atas nama waktu salah satunya adalah firman Allah SWT dalamQ.S. Al-Asr ayat 1-3 yang berbunyi:

وَالْعَصْرِۙ – ١ اِنَّ الْاِنْسَانَ لَفِيْ خُسْرٍۙ – ٢ اِلَّا الَّذِيْنَ اٰمَنُوْا وَعَمِلُوا الصّٰلِحٰتِ وَتَوَاصَوْا بِالْحَقِّ ۙ وَتَوَاصَوْا بِالصَّبْرِ ع – ٣

Artinya:

"Demi masa (1), sungguh, manusia berada dalam kerugian (2), kecuali orang-orang yang beriman dan mengerjakan kebajikan serta saling menasihati untuk kebenaran dan saling menasihati untuk kesabaran (3)".

Ayat di atas menunjukkan bahwa waktu memegang peranan penting dalam kehidupan manusia. Kehidupan di dunia ini memiliki waktu yang singkat. Dengan begitu itu manusia harus memanfaatkan waktu dengan sebaik mungkin untuk berbuat baik dan menasehati dalam kebaikan. Jika kita tidak berbuat baik selama hidup maka kita akan masuk dalam golongan yang merugi. Manusia selama di dunia harus mempergunakan waktu sebaik mungkin untuk memperbanyak amal ibadah dan maupun berbuat kebaikan. Dengan begitu itu seorang pekerja yang telah memegang amanah yang telah disepakati yaitu dengan memiliki waktu-waktu tertentu dalam bekerja dan ia digaji dalam waktu-waktu tersebut seharusnya menggunakan waktu-waktu tersebut dengan seefisien dan seefektif mungkin. untuk menunaikan amanah yang telah disepakati. Sebagaimana Allah Ta'ala berfirman dalam QS. Al-Nisa (4): 58 yang berbunyi:

اِنَّ اللّٰهَ يَأْمُرُكُمْ اَنْ تُؤَدُّوا الْاَمٰنٰتِ اِلٰٓى اَهْلِهَاۙ

Artinya:

"Sesungguhnya Allah memerintahkan kepada kalian untuk menunaikan amanat kepada yang berhak"

Ayat diatas dapat disimpulkan bahwa terutama di masa pandemi ini beberapa perusahaan menerapkan sistem WFH untuk membantu pemerintah dalam mencegah penyebaran COVID-19. Sehingga mengharuskan banyak pekerja bekerja

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dari rumah. Pekerja haruslah dapat memanfaatkan waktu dengan seefisien dan seefektif mungkin meskipun bekerja dari rumah. Teknologi digital menjadi *backbone* untuk para pekerja yang sedang menjalankan pekerjaanya dari rumah. Aplikasi *capture screen* monitor selain membantu karyawan atau atasan dalam memantau karyawannya aplikasi ini juga dapat membantu para karyawan terhindar dari korupsi waktu yang dapat berakibat terhadap keberkahan dari harta yang diperoleh.

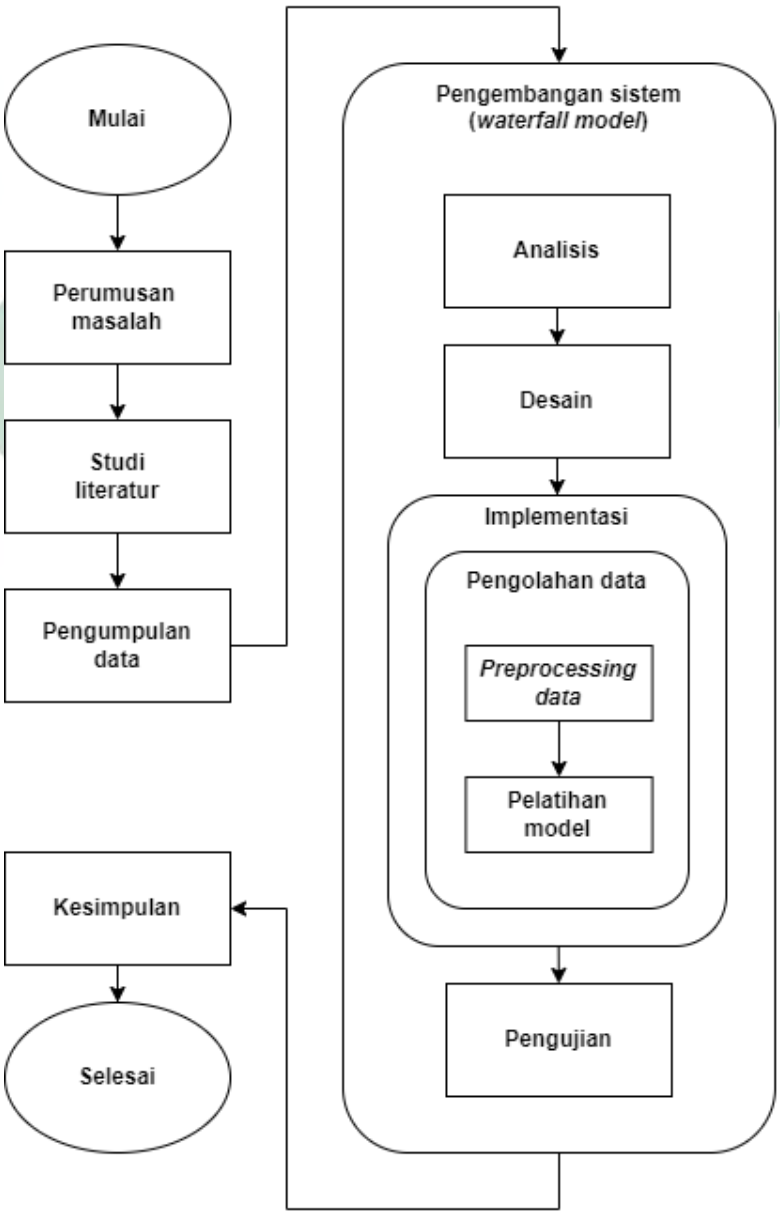
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Dalam membantu menyusun penelitian supaya lebih mudah dalam memahami dan menyampaikan informasi dan langkah-langkah yang dilakukan, maka alur penelitian dipresentasikan ke dalam *flow diagram* yang terdapat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada Gambar 3.1 dapat dijelaskan alur penelitian dimulai dari perumusan masalah, studi literatur, pengolahan data, pengambangan sistem dan evaluasi sistem. Berikut penjelasan dari masing-masing setiap prosesnya.

3.1.1 Perumusan Masalah

Perumusan masalah yang sebagaiman sudah tertera pada bab pertama yang memiliki dua poin penting yaitu cara mengembangakn aplikasi *screen monitor duration* karyawan saat bekerja dan cara melakukan pengujian fungsi aplikasi. Penelitian ini menjawab rumusan masalah tersebut dengan tujuan untuk mengetahui tentang langkah-langkah pengembangan aplikasi dengan metode SVM dan mengetahui nilai presentase fungsi aplikasi dapat berjalan sesuai kebutuhan.

3.1.2 Studi Literatur

Penelitian ini menggunakan studi literatur dalam memahami konsep-konsep dan variabel-variabel yang relevan pada penelitian ini. Studi literatur didapatkan dari jurnal-jurnal yang berkaitan dengan penelitian dan buku-buku yang relevan dengan konsep-konsep yang ada pada penelitian.

3.1.3 Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan *tracking* akun karyawan ketika menjalankan aplikasi ini kemudian berelasi pada suatu *project* dan *task* yang akan dikerjakan. Dari proses tersebut akan menghasilkan data-data seperti pada Tabel 3.1.

Tabel 3.1 Variabel-variabel pada data *screen monitor duration* karyawan

Nama variabel	Keterangan
Account ID	Akun yang dipakai karyawan saat menjalankan aplikasi
Project ID	Proyek yang dipakai karyawan saat akan melakukan *screen monitor duration*
Task ID	Tugas yang akan dikerjakan karyawan dan berkaitan dengan variabel *work in progress* serta *standart minute value* dalam dataset
Process ID	Proses-proses yang dijalankan karyawan seperti aplikasi apa saja yang digunakan karyawan
Tracking ID	Menyimpan aktivitas waktu dan pekerjaan apa yang dilakukan karyawan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada Tabel 3.1 dapat disimpulkan bahwa untuk mengembangkan aplikasi *capture screen* monitor dalam mengukur produktifitas karyawan membutuhkan variabel-variabel yaitu *account id* sebagai akun karyawan, *project id* sebagai wadah semua pekerjaan, *task id* dan *tracking id* sebagai data acuan dalam klasifikasi.

3.1.4 Pengembangan Sistem

Pengembangan perangkat lunak aplikasi *screen monitor duration* ini menggunakan model pengembangan *waterfall model* dengan langkah-langkah sebagai berikut:

1. Analisis

   Fungsi utama aplikasi ini adalah untuk menentukan produktifitas karyawan selama bekerja. Analisis kebutuhan dilakukan dengan cara studi literatur dengan menggetahui masalah-masalah yang ada dan informasi-informasi yang digunakan untuk memecahkan masalah tersebut. Setelah itu, infomasi-informasi tersebut akan diolah menjadi kebutuhan-kebutuhan aplikasi.

   Dari pernyataan diatas dapat disimpulkan bahwa pada tahap analisis akan dilakukan analisis kebutuhan *fungsional* dan kebutuhan *non fungsional* serta spesifikasi kebutuhan perangkat lunak dan perangkat keras.

2. Desain

   Tahapan desain adalah melakukan rencana penggambaran sistem yang ada pada aplikasi yang dikembangkan yang diperoleh dari tahap analisis. Pada tahap ini dilakukan desain UML, dan desain basis data.

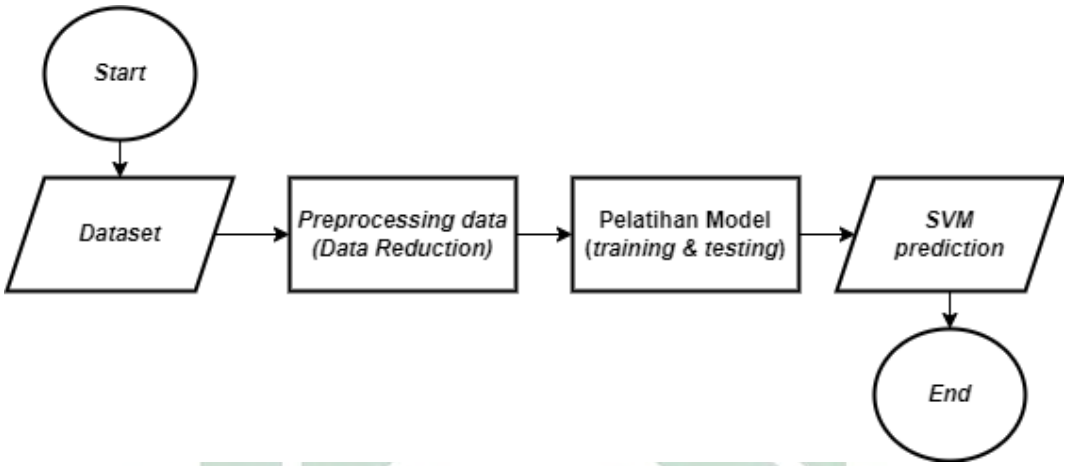
3. Implementasi

   Pada tahap implementasi adalah melakukan eksekusi dari desain sistem yang sudah dilakukan pada tahap sebelumnya. Implementasi ini dibagi menjadi dua yaitu implementasi *Support Vector Machine* (SVM) untuk klasifikasi dan implementasi pengembangan aplikasi *desktop* sebagai *client* dan aplikasi *website* sebagai *server*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

A. Implementasi *Support Vector Machine* (SVM)

Implementasi SVM dilakukan untuk mendapatkan klasifikasi prediksi dengan tahapan-tahapan pada gambar 3.2



Gambar 3.2 Tahapan Klasifikasi SVM

1. *Preprocessing data*

Dataset yang digunakan untuk melakukan klasifikasi *screen monitor duration* untuk menghasilkan kinerja karyawan adalah dataset dari website *UCI Machine Learning Repository* dengan judul "*Productivity Prediction of Garment Employees Dataset* (https://archive.ics.uci.edu/ml/datasets/)" dengan total data 1197 data yang terdiri dari 15 kolom variabel. Pada tahap *preprocessing data* ini dilakukan agar dapat mengenali karakter data yang telah dikumpulkan sehingga mendapatkan hasil prediksi yang lebih akurat. Dalam tahap ini data-data yang digunakan akan diberi variabel-variabel seperti tabel 3.2

Tabel 3.2 Variabel Dataset

Variabel	Nilai	Keterangan
no_of_workers	Sejumlah nilai angka bulat lebih besar dari 0 ($nilai > 0$)	Jumlah karyawan di setiap tim
target_productivity	Sejumlah nilai angka berkisar dari 0-1 ($nilai \leq 1$ & $nilai \geq 0$)	Target produktivitas yang ditetapkan oleh Manager untuk setiap hari.
smv	Sejumlah nilai angka lebih besar dari 0 ($nilai > 0$)	*Standard Minute Value* adalah alokasi waktu untuk suatu tugas

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

wip	Sejumlah nilai angka bulat lebih besar dari 0 ($nilai \geq 0$)	*Work in progress* adalah jumlah tugas yang belum selesai dikerjakan
over_time	Sejumlah nilai angka bulat lebih besar dari 0 ($nilai \geq 0$)	Mewakili jumlah *overtime* oleh setiap tim dalam menit
actual_productivity	Sejumlah angka bernilai 0 atau 1	aktual dari produktivitas yang dihasilkan oleh karyawan yang bernilai 1 *(productive)* atau bernilai 0 *(unproductive)*

2. Pelatihan model

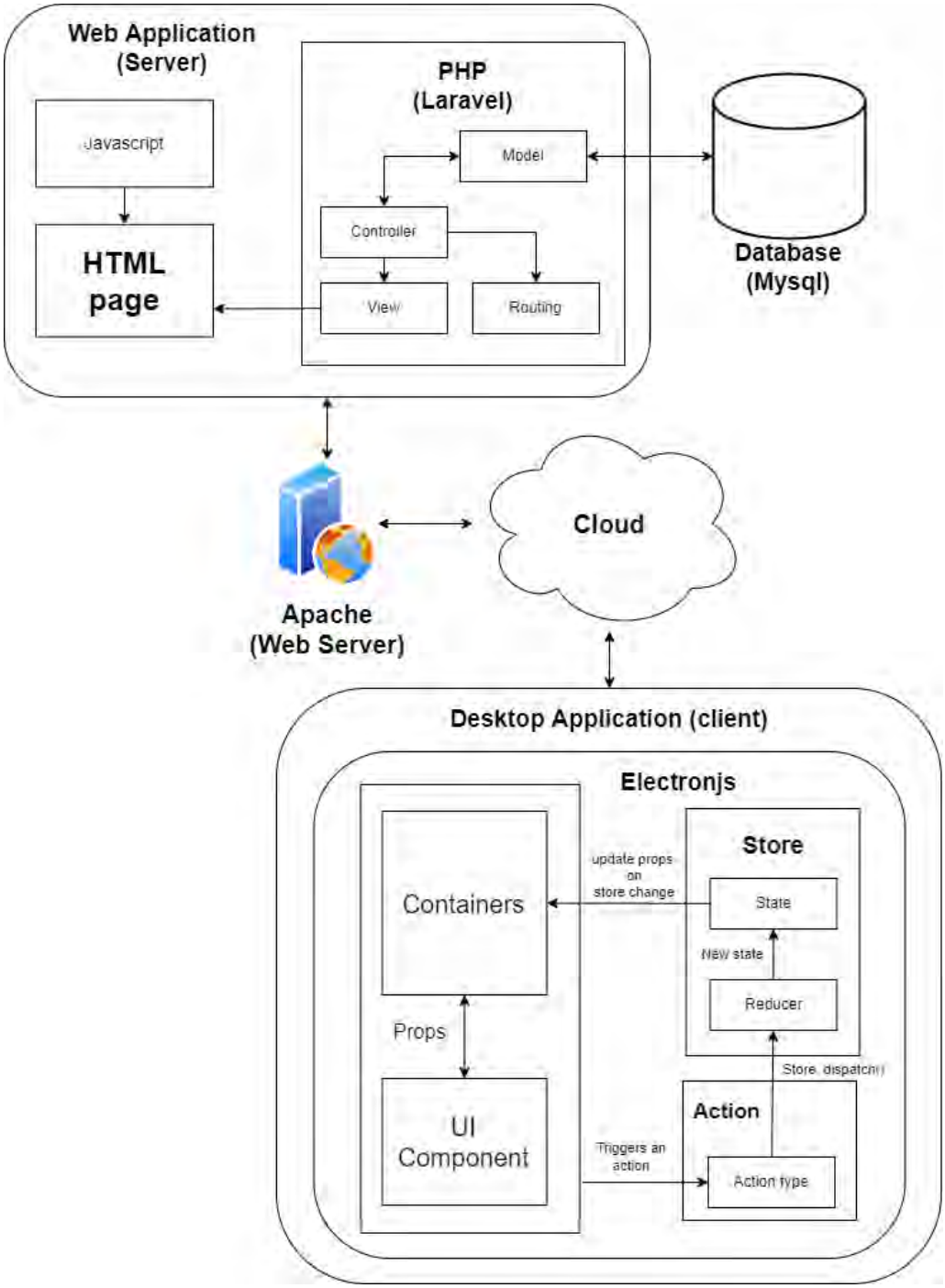
Dataset yang sudah melalui *preprocessing data,* maka dataset tersebut sudah siap untuk proses pelatihan model. Proses pelatihan model ini terdiri dari proses *training* dan proses *testing* dengan menggunakan pembagian proporsi ratio dataset dengan *data training* dan *data testing.* Proses pelatihan model ini menggunakan metode *Support Vector Machine* (SVM) dan metode pembanding *Support Vector Regression* (SVR).

B. Implementasi Pengembangan Aplikasi

Pada aplikasi *desktop* dibangun menggunakan *framework electronjs* dengan bahasa pemrograman *javascript*. Selanjutnya pada aplikasi *website* dibangun menggunakan *framework Laravel*.

Pada masing-masing aplikasi akan saling terhubung satu sama lain dengan interaksi pertukaran data. Interaksi pertukaran data ini menggunakan *web service RESTful API* dengan komunikasi data berformat *JavaScript Object Notation* (JSON). Dalam komunikasi masing-masing aplikasi dapat dilihat arsitektur aplikasinya pada gambar 3.2

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 3.3 Arsitektur Aplikasi

4. Pengujian

Pada tahap terakhir pengujian perangkat lunak dengan menguji *prototype* yang sudah dibuat pada tahap sebelumnya. Pada pengujian ini dibagi menjadi tiga kategori sebagai berikut:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

a. Pengujian *fungsional*

Pengujian *fungsional* memiliki fitur-fitur yang akan diuji. Fitur-fitur ini berada pada aplikasi pengguna dengan rangkaian fitur seperti pada Tabel 3.3.

Tabel 3.3 Fitur yang akan di uji

Fitur	Proses	*Test* ID
Login	Validasi email dan password	A01
	Feedback login berhasil dan gagal	A02
Register	Validasi form inputan	B01
	Feedback register berhasil dan gagal	B02
Tracking	*Time running*	C01
	Application running	C02
	Screen shoot	C03
Result Work	*Show tracking report*	D01
	Show productivity percentage	D02

Pada Tabel 3.3 fitur-fitur yang akan diuji merupakan fitur utama, kemudian untuk fitur-fitur selain fitur utama tidak masuk dalam pengujian. Setelah itu pengujian ini akan menggunakan skenario pengujian yang terdapat pada Tabel 3.4.

Tabel 3.4 Skenario Pengujian

Fitur	*Test Case ID*	*Test Scenario*	Persyaratan
Login	A01	*Login User*	Login menggunakan email dan password yang sudah ada pada database
			Email harus termasuk email
Register	B01	*Register User*	Registrasi menggunakan nama, email, password dan confirm password yang terisi
			Password harus minimal 8 karakter
			Email tidak boleh ada yang sama

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Fitur	*Test Case ID*	*Test Scenario*	Persyaratan
Capture screen	C01	*Time Running*	Saat menekan tombol "Mulai", waktu akan otomatis berjalan
			Waktu akan berjalan mulai dari detik pertama
	C02	*Application running*	Saat *time running* berjalan, *user* membuka aplikasi apa saja yang ada di komputer akan di rekam oleh sistem
	C03	*Interval Screen Shoot*	Saat *time running* berjalan, sistem akan melakukan *screen shoot* otomatis dengan *interval* tertentu
Result Work	D01	*Show capture screen report*	Saat menekan tombol "*Show capture screen report*", akan menuju halaman *capture screen report*
	D02	*Show Productivity percentage*	Saat menekan tombol "*Show productivity percentage*", akan menuju halaman presentase produktivitas

b. Pengujian *Root Mean Square Error* (RMSE)

Pengujian RMSE ini dilakukan setelah melakukan pelatihan model. Model yang sudah dilatih akan dihitung nilai RMSE dengan rumus:

$$\text{RMSE}=\sqrt{\frac{\sum(\text{Aktual-Predikat})^2}{\text{n}}}$$

Hasil perhitungan nilai RMSE dari metode SVM dan SVR akan dibandingkan dengan nilai terendah yang menunjukkan bahwa variasi nilai yang dihasilkan oleh suatu model tersebut semakin dekat dengan nilai yang diprediksi.

c. Pengujian performa

Menguji performa terbaik yang digunakan pada pelatihan model oleh metode SVM dan SVR. Penggujian performa menguji dengan menentukan parameter nilai *cost* dan *gamma* terbaik sebagai parameter untuk pelatihan model. Hasil pengujian

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ini digunakan dalam implementasi SVM dan SVR dengan menggunakan parameter *cost* dan *gamma* performa terbaik.

3.2 Tempat dan Waktu Penelitian

Tempat dan waktu penelitian ini didapatkan melalui pertimbangan pokok masalah dan waktu untuk melakukan penelitian.

3.2.1 Tempat Penelitian

Penelitian ini dilakukan pada perusahaan-perusahaan yang memiliki kebutuhan akan monitoring karyawan terlebih perusahaan yang menerapkan sistem WFH.

3.2.2 Waktu Penelitian

Penelitian ini dilakukan dalam kurun waktu 18 minggu mulai dari bulan July minggu ke-3 sampai akhir bulan November dari kegiatan perumusan masalah, studi literatur, pengumpulan data, pengolahan data hingga pengembangan sistem.

Pada kegiatan perumusan masalah dan studi literatur akan dilakukan selama dua minggu, kemudian pengumpulan dan pengolahan data dilakukan selama empat minggu dan terakhir pengembangan sistem dilakukan selama 6 minggu. Detail waktu kegiatan penelitian dapat dilihat pada tabel 3.5

Tabel 3.5 Waktu Penelitian

Keterangan	July		Agustus				September				Oktober				November			
	1	2	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Perumusan masalah																		
Studi literatur																		
Pengumpulan data																		
Pengolahan data																		
Pengembangan sistem																		

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV
HASIL DAN PEMBAHASAN

Pengembangan aplikasi menggunakan *Waterfall Model* sebagai tahapan pengembangan aplikasi yang terdiri dari empat tahapan yaitu analisis, desain, implementasi dan pengujian.

4.1 Analisis

Pada tahap analisis melakukan pengumpulan data-data terkait pengembangan aplikasi hingga menghasilkan keluaran untuk tahapan desain. Analisis dalam pengembangan aplikasi ini meliputi analisis kebutuhan *fungsional* dan *non fungsional,* serta analisis kebutuhan perangkat lunak dan perangkat keras.

4.1.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini mencangkup kebutuhan-kebutuhan utama yang harus ada pada aplikasi. Kebutuhan fungsional aplikasi dapat dilihat pada tabel 4.1.

Tabel 4.1 Kebutuhan fungsional

No	Kebutuhan Fungsional
1	Aplikasi harus dapat melakukan *login* dan *register* user
2	Aplikasi harus dapat mengidentifikasi *role user*
3	Aplikasi harus dapat memanajemen proyek
4	Aplikasi harus dapat memanajemen tugas yang berkaitan dengan karyawan
5	Aplikasi harus dapat melakukan *capture application running* saat *start time*
6	Aplikasi harus dapat melakukan *screen shoot* otomatis saat *start time*
7	Aplikasi harus dapat melakukan klasifikasi kinerja karyawan
8	Aplikasi harus dapat menampilkan *report* kinerja karyawan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada tabel 4.1 dijelaskan bahwa kebutuhan utama aplikasi ini ada delapan dimana kebutuhan-kebutuhan ini harus ada pada aplikasi dengan dibuktikan oleh pengujian aplikasi.

4.1.2 Analisis Kebutuhan *non Fungsional*

Kebutuhan *non fungsional* ini meliputi kebutuhan-kebutuhan sekunder artinya tidak harus ada pada aplikasi. Kebutuhan ini berkaitan dengan aplikasi yang dapat dilihat pada tabel 4.2.

Tabel 4.2 Kebutuhan *non Fungsional*

No	Kebutuhan *non Fungsional*
1	Aplikasi harus dapat menampilkan aktivitas user dengan *capture screen* secara *real time*
2	Aplikasi harus dapat menampilkan informasi *user* secara *real time*
3	Gambar *screen shoot* otomatis harus berkualitas sangat bagus
4	Ukuran *file* aplikasi *desktop* tidak boleh lebih dari 10 *Mega Byte* (MB)

Pada tabel 4.2 kebutuhan sekunder yang dimiliki apikasi ini ada empat dimana semuanya ini berhubungan dengan kualitas aplikasi yang dikembangkan. Kebutuhan *non fungsional* ini tidak harus diterapkan tetapi akan menjadi nilai tambahan jika diterapkan.

4.1.3 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak ini digunakan dalam membangun aplikasi ini. Perangkat lunak yang digunakan harus memiliki spesifikasi versi minimal seperti pada tabel 4.3.

Tabel 4.3 Kebutuhan perangkat lunak

Perangkat lunak	Versi
Sistem operasi	*Windows 7*
Visual studio code	Versi 1.54.3
XAMPP	Versi 3.2.4
Composer	Versi 2.0.11
Node js	Versi 10

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada tabel 4.3 kebutuhan perangkat lunak yang digunakan dengan sistem operasi *windows* digunakan berdasarkan pengembangan aplikasi ini dikhusus kan pada sistem operasi *windows*. *Visual studio code* digunakan sebagai editor terbaik karena dapat terintegrasi banyak *ekstensi* yang mendukung pemrograman. *Composer* dan *node js* digunakan sebagai *framework* dalam pengembangan aplikasi yang menggunakan *laraval* dan *electronjs*.

4.1.4 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yang digunakan dalam membangun aplikasi ini harus memiliki spesifikasi minimal seperti pada tabel 4.4.

Tabel 4.4 Kebutuhan perangkat keras

Perangkat keras	Versi
Processor	Setara dengan *Intel Core i3 2100*
RAM	2 *Giga Byte* (GB)
Hardisk	320 *Giga Byte* (GB)
VGA	*Intel HD 620/615*

Pemilihan perangkat keras dengan spesifikasi minimal pada tabel 4.4 didasarkan pada kecepatan optimal dalam penggunaannya dengan biaya tidak terlalu mahal dan mengikuti spesifikasi perangkat keras minimum pada umumnya.

4.2 Desain

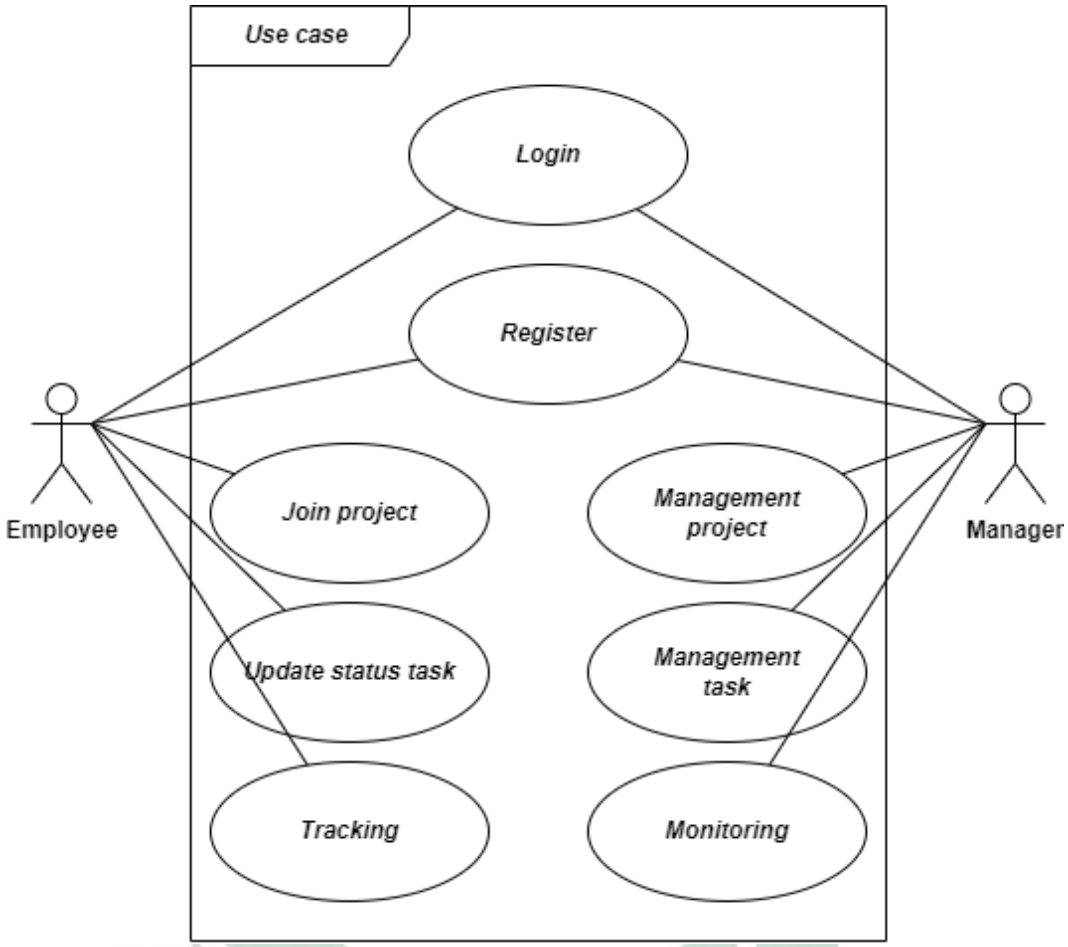
Pada tahap desain melakukan penggambaran-penggambaran pada aplikasi yang dikembangkan hingga menghasilkan keluaran untuk tahap implementasi. Desain dalam pengembangan aplikasi ini meliputi desain UML, desain *user interface* dan desain *database.*

4.2.1 Desain UML

Desain UML diagram ini digunakan untuk desain sistem yang terdiri dari *use case diagram, activity diagram, sequence diagram* dan *class diagram.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. *Use case diagram*



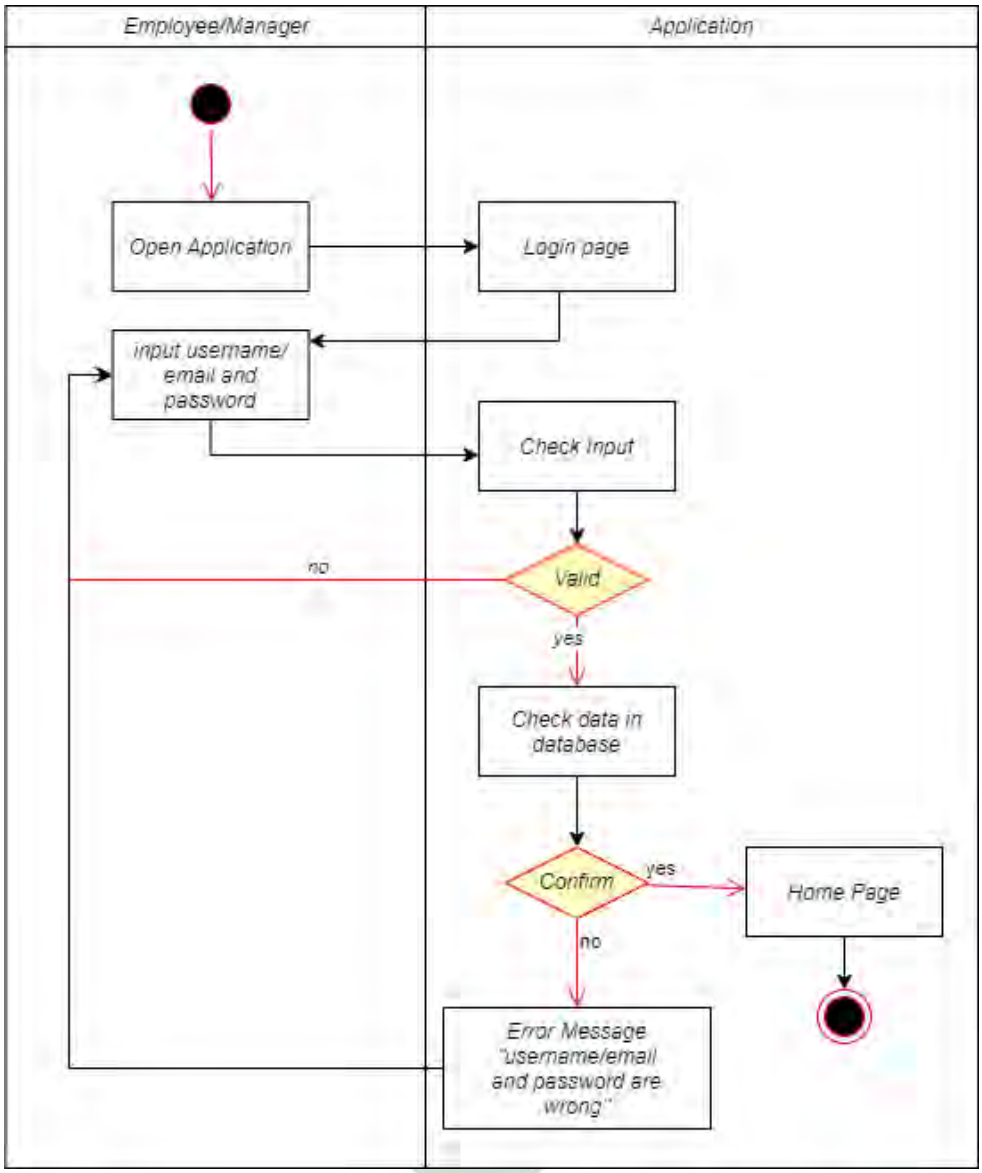
Gambar 4.1 *Usecase Diagram* Penelitian ini

Pada gambar 4.1 menjelaskan sebuah kegiatan yang dilakukan oleh aktor. Aktor dalam aplikasi ini yaitu *Employee* dan *Manager* yang memiliki aktivitas yang berbeda. *Employee* memiliki kegiatan pada sistem antara lain yaitu *register, login, join project, update status task,* dan *tracking.* Manager memiliki kegiatan pada sistem antara lain *register, login, management project, management task,* dan *monitoring.*

Pada kegiatan *register* melakukan pendaftaran akun yang akan digunakan dalam aplikasi untuk masuk pada aplikasi. Pada menu *management project*, *Manager* membuat sebuah proyek yang mana *Employee* akan bergabung pada proyek tersebut. Setelah itu *Manager* akan membuat *task* untuk dikerjakan *Employee*, *Employee* akan melakukan *tracking* untuk mengerjakan *task* dan setelah selesai akan mengubah status *task.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

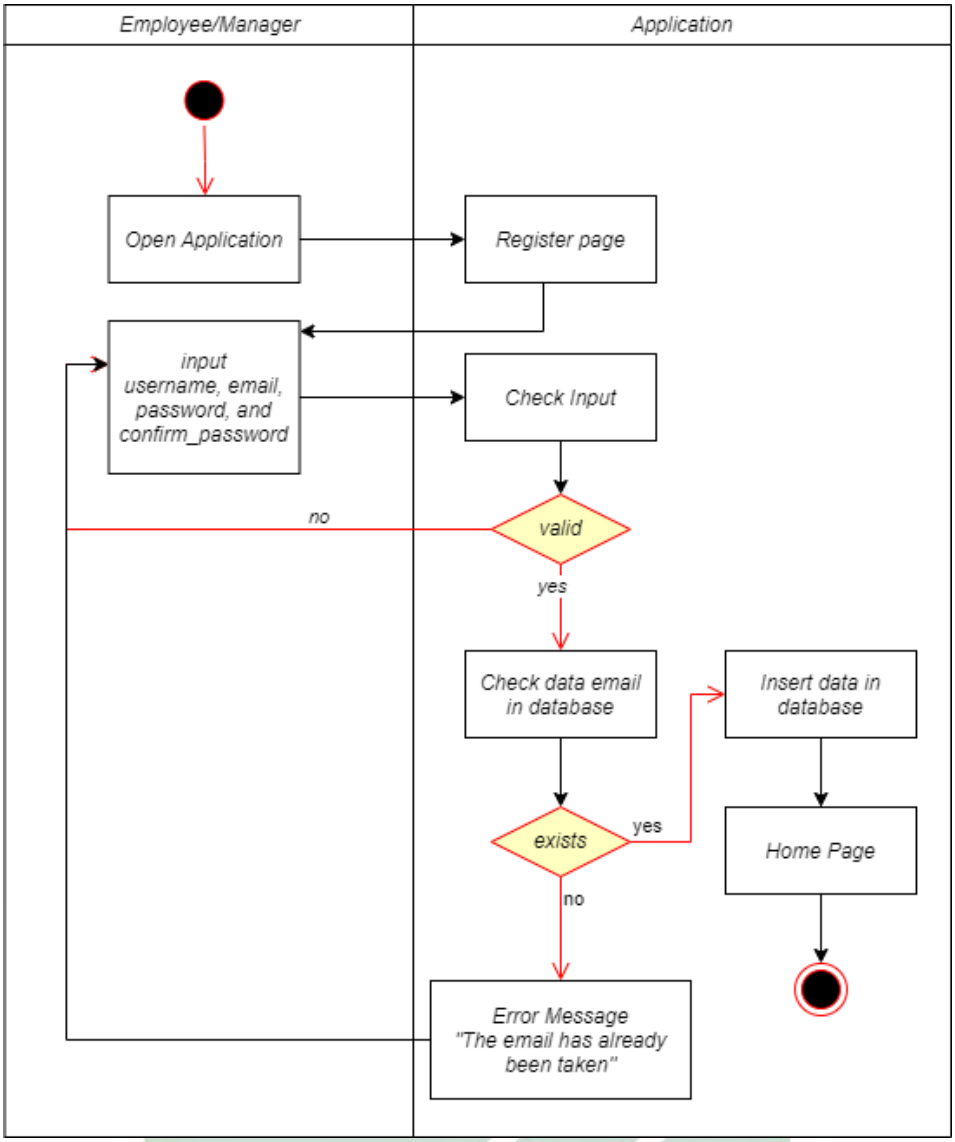
2. *Activity diagram*



Gambar 4.2 *Activity Diagram Login*

Pada gambar 4.2 digambarkan sebuah aktivitas *login* yang dilakukan oleh *employee* atau *manager*. Aktivitas pertama yang dilakukan yaitu membuka aplikasi kemudian dari aplikasi akan menampilkan halaman *login.* Selanjutnya pengguna memasukan *username/email* dan *password,* lalu aplikasi akan melakukan *check inputan,* jika *valid* aplikasi akan *check data* pada database jika *invalid* aplikasi akan mengembalikan ke halaman *login.* Pada aktivitas *check data in database,* jika datanya ada aplikasi akan menampilkan halaman *home,* jika datanya tidak ada aplikasi akan memunculkan pesan *error* dan mengembalikan pada halaman *login.*

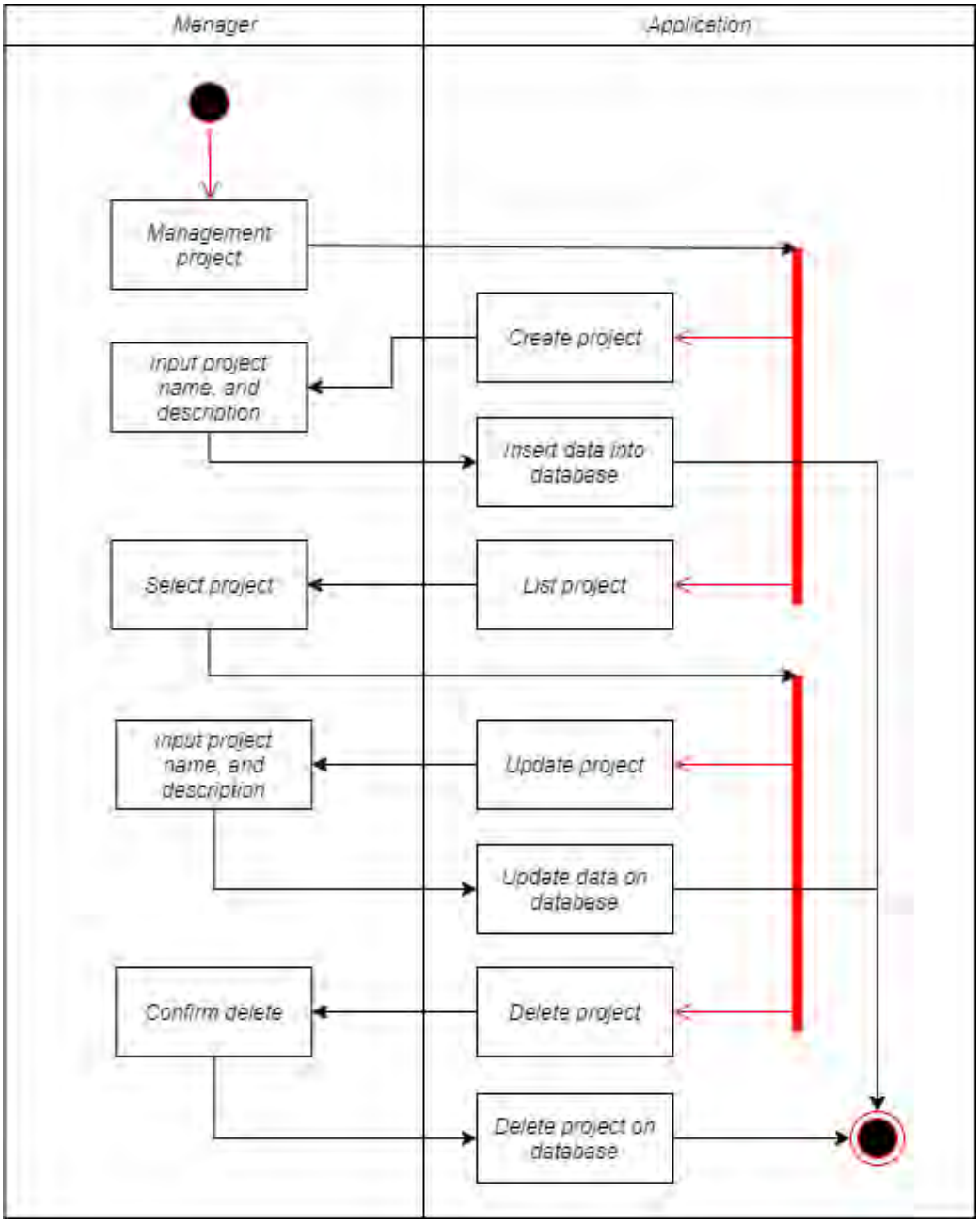
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.3 *Activity Diagram Register*

Pada gambar 4.3 digambarkan sebuah aktivitas *register* yang dilakukan oleh *employee* atau *manager*. Aktivitas pertama yang dilakukan yaitu membuka aplikasi kemudian dari aplikasi akan menampilkan halaman *register.* Selanjutnya pengguna memasukan *username, email, password* dan *confirm password,* lalu aplikasi akan melakukan *check inputan,* jika *valid* aplikasi akan *check data email* pada database jika *invalid* aplikasi akan mengembalikan he halaman *register.* Pada aktivitas *check data email in database,* jika datanya tidak ada aplikasi akan menambahkan data lalu menampilkan halaman *home,* jika datanya ada aplikasi akan memunculkan pesan *error* dan mengembalikan pada halaman *register.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

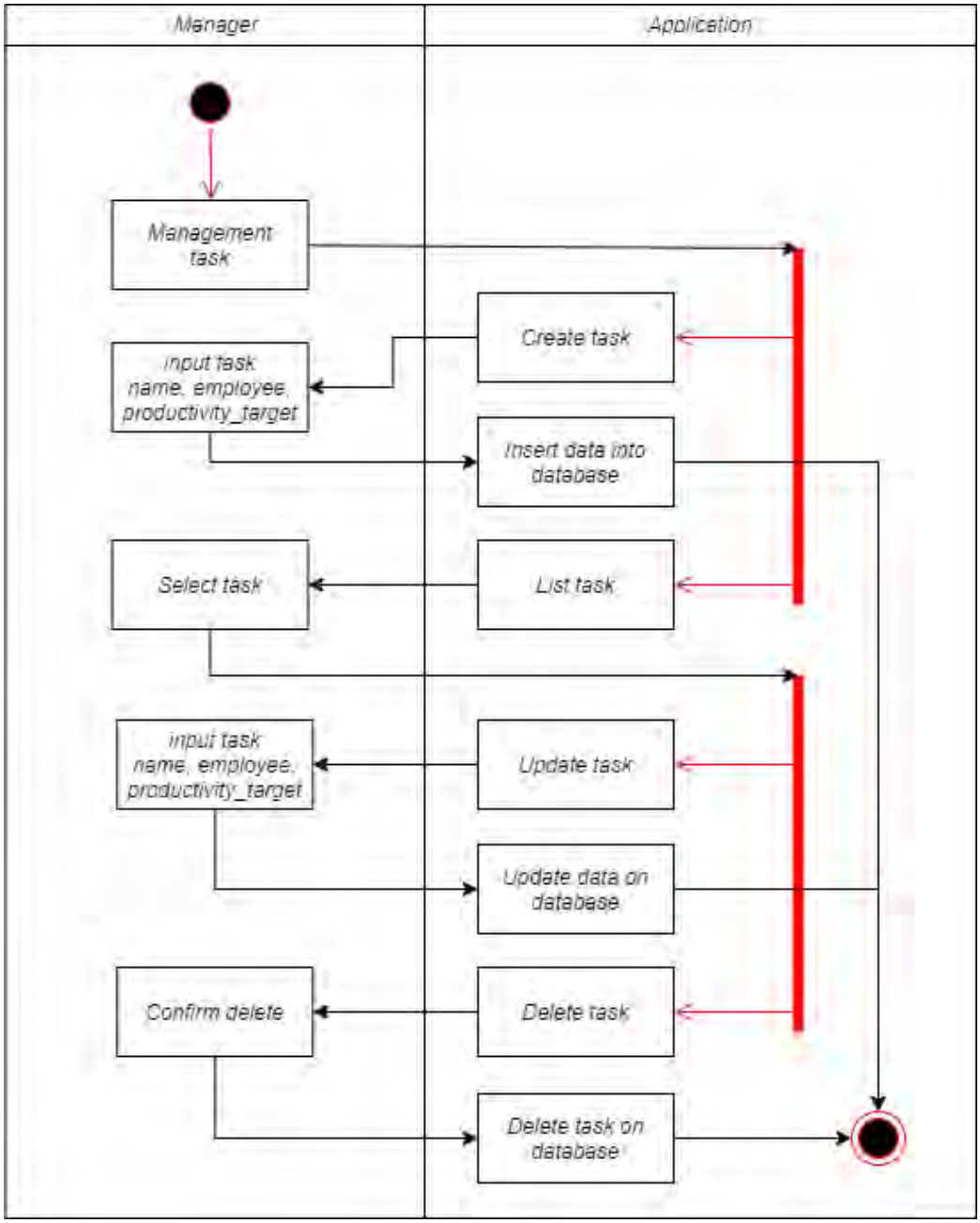


Gambar 4.4 *Activity Diagram Management Project*

Pada gambar 4.4 digambarkan sebuah aktivitas *management project* yang dilakukan oleh *manager*. Aktivitas pertama yang dilakukan yaitu memilih menu *management project* kemudian dari aplikasi akan menampilkan dua opsi yaitu *create project* dan *project list.* Pada aktivitas *create project, manager* akan memasukan nama proyek dan deskripsi pada *form* lalu aplikasi akan menyimpan data tersebut dalam database.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada *project list, manager* akan memilih proyek lalu dari proyek tersebut memiliki dua opsi lagi yaitu *update project* atau *delete project.* Pada *update project, manager* akan mengisi *form* lalu aplikasi akan merubah data sesuai masukan dari *manager.* Kemudian pada aktivitas *delete project, manager* akan mengkonfirmasi terlebih dahulu setelah itu aplikasi akan menghapus data yang ada dalam *database.*

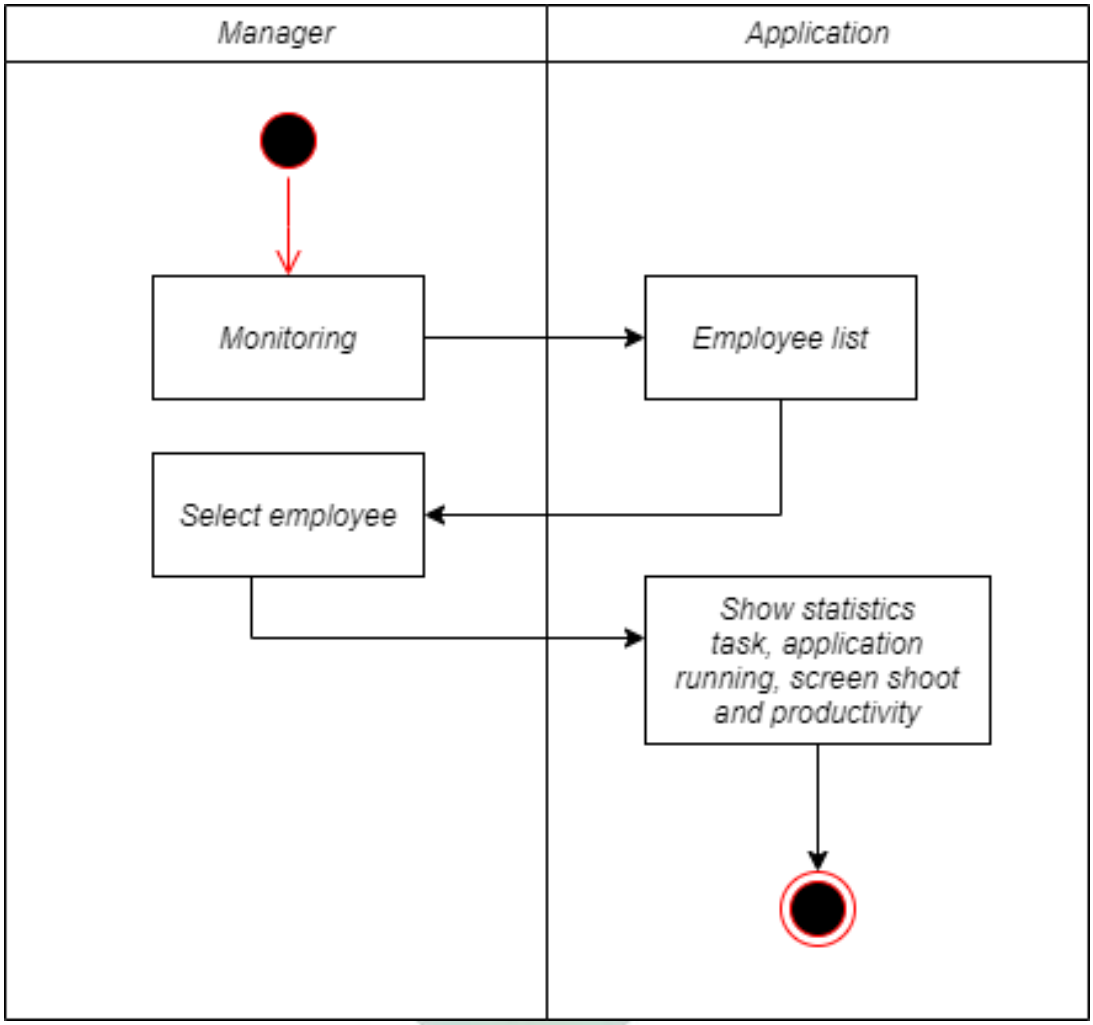


Gambar 4.5 *Activity Diagram Management Task*

Pada gambar 4.5 digambarkan sebuah aktivitas *management task* yang dilakukan oleh *manager*. Pada aktivitas ini semua aktivitas proses hampir sama

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

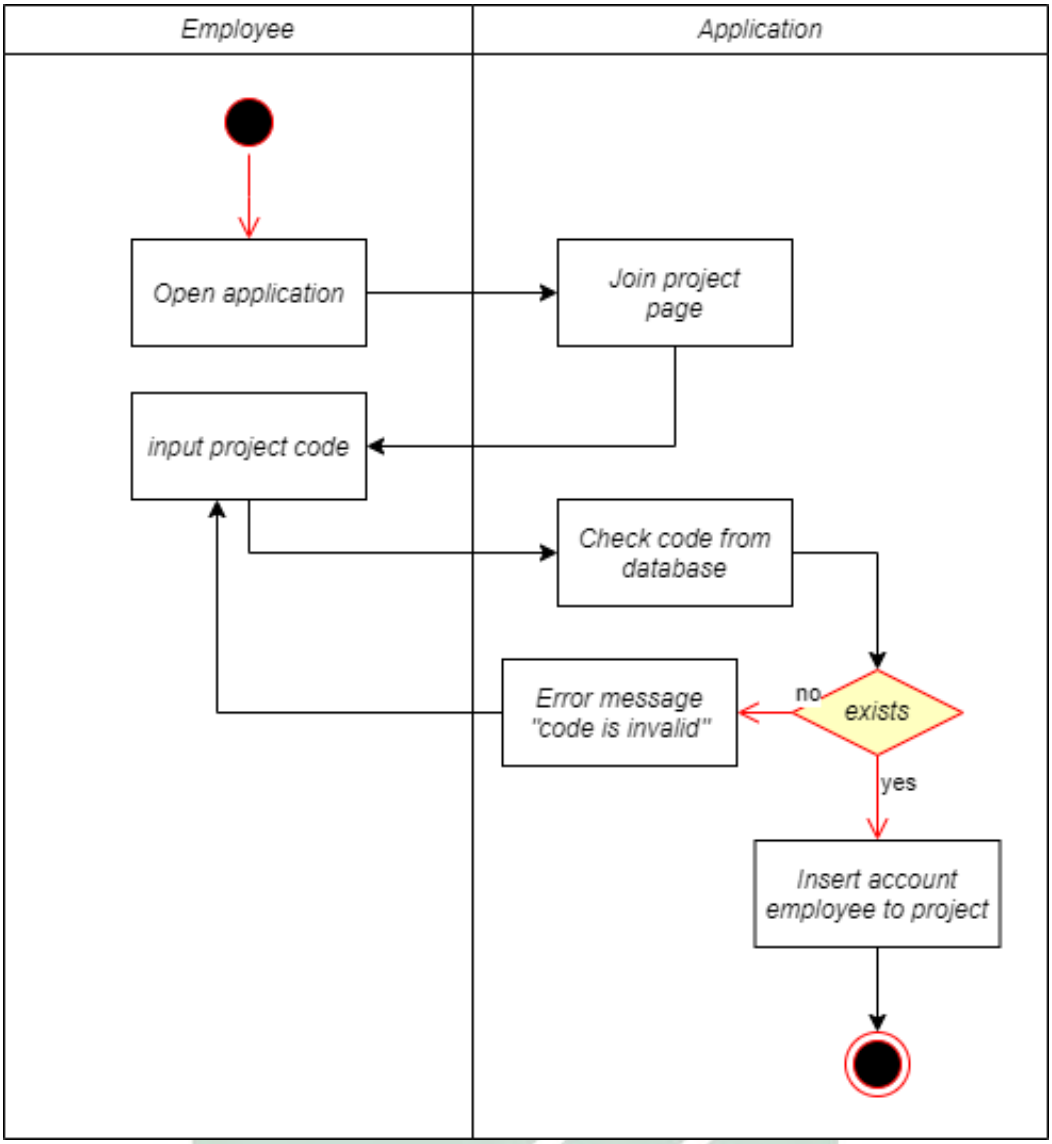
dengan *management project.* Perbedaan pada manament *task* yaitu pada bagian form memiliki atribut nama *task, employee* pada *task* dan target produktivitas karyawan.



Gambar 4.6 *Activity Diagram Monitoring Employee*

Pada gambar 4.6 digambarkan sebuah aktivitas *monitoring employee* yang dilakukan oleh *manager*. Aktivitas pertama yang dilakukan yaitu memilih menu *monitoring* lalu pada aplikasi akan menampilkan *employee list.* Selanjutnya *manager* melakukan *select project* lalu pada aplikasi akan menampilkan detail statisistik *task, application open, screen shoot* dan *productivity employee.*

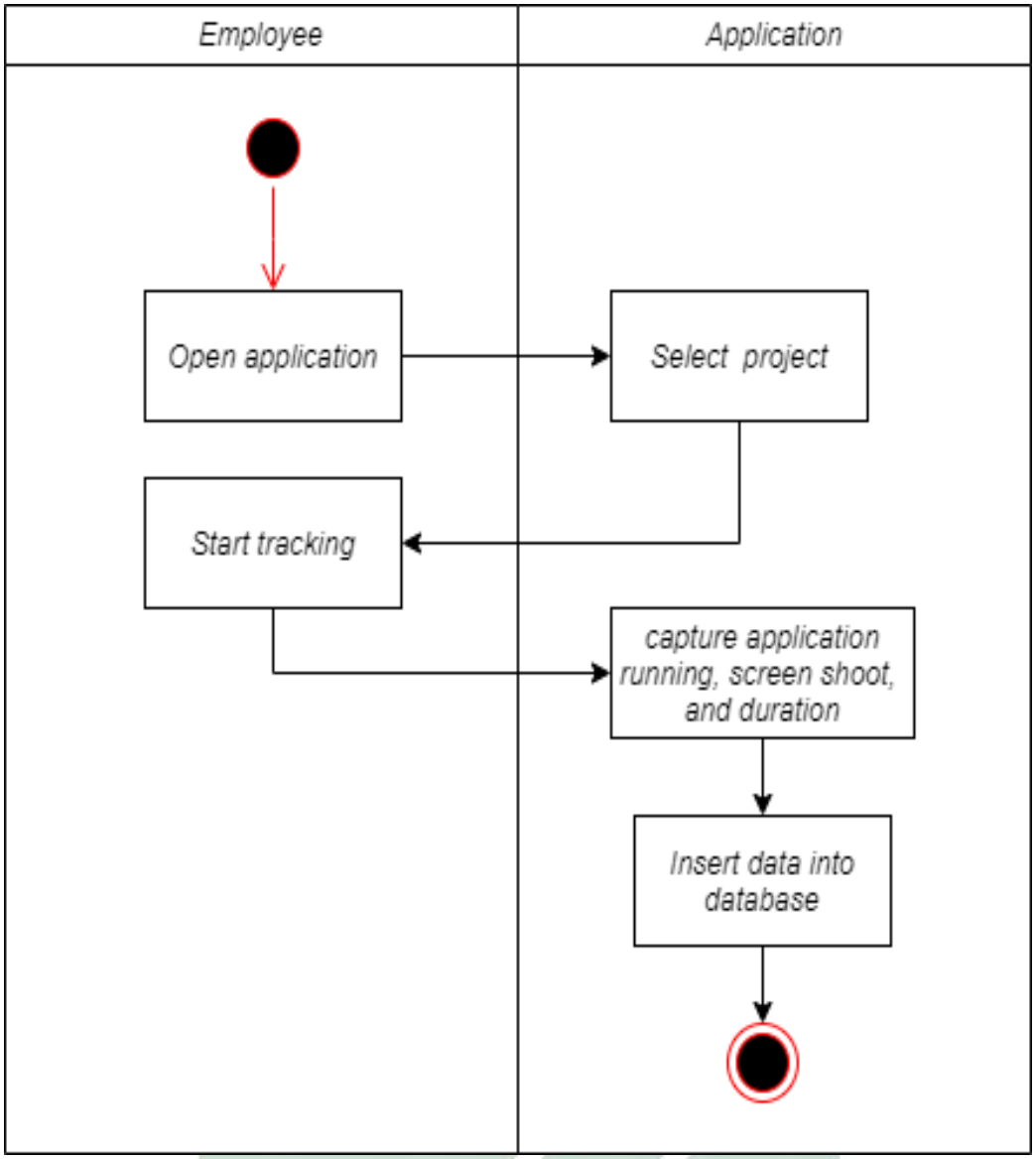
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.7 *Activity Diagram Join Project*

Pada gambar 4.7 digambarkan sebuah aktivitas *join project* yang dilakukan oleh *employee*. Aktivitas pertama yang dilakukan yaitu membuka aplikasi lalu pada aplikasi akan menampilkan halaman *join project.* Kemudian *employee* memasukan kode proyek lalu aplikasi akan melakukan *check code from databse,* jika kodenya sesuai, aplikasi akan melakukan *insert account employee to project*, jika tidak sesuai, aplikasi akan memunculkan pesan kesalahan dan mengembalikan pada halaman *join project.*

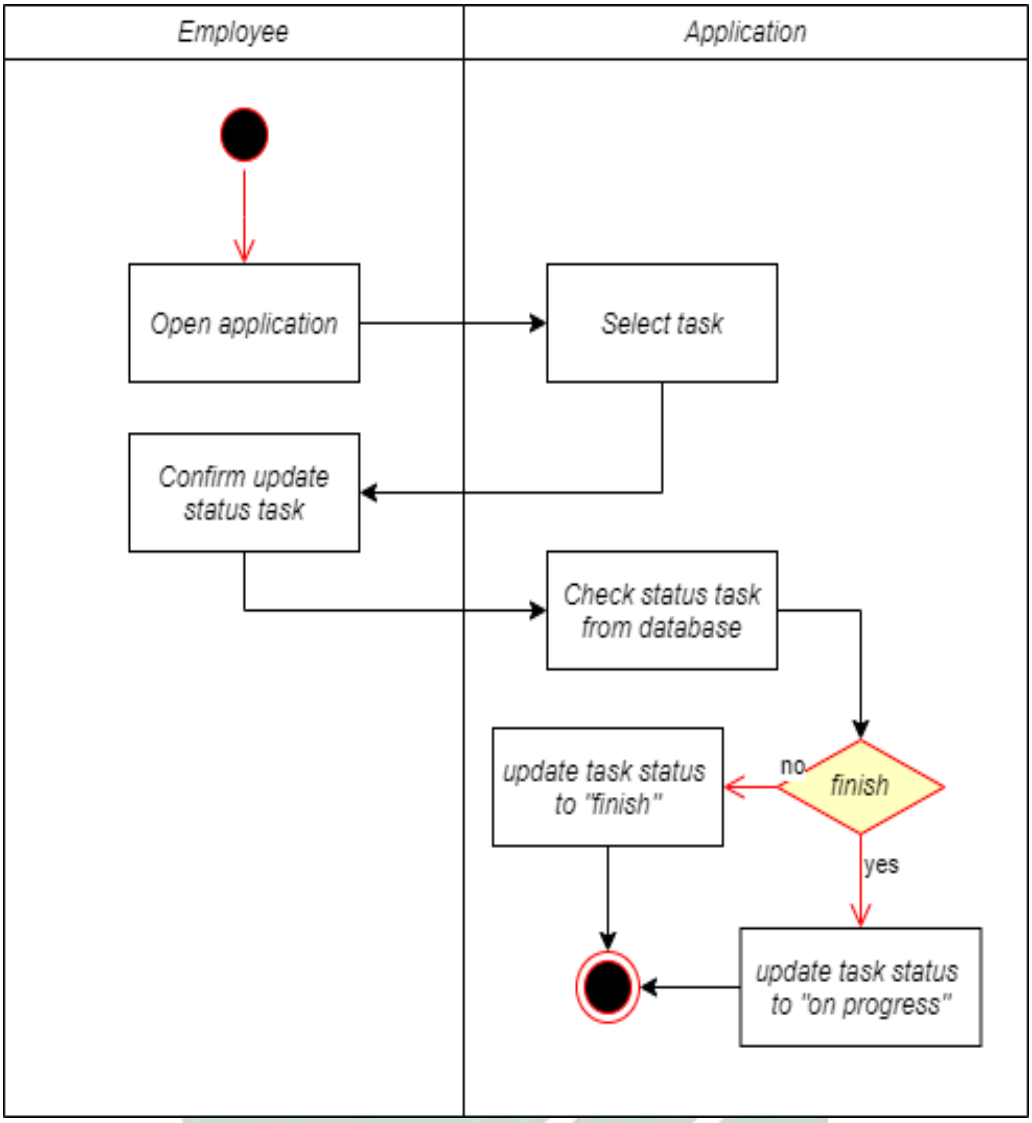
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.8 *Activity Diagram Tracking*

Pada gambar 4.8 digambarkan sebuah aktivitas *tracking* yang dilakukan oleh *employee*. Pada aktivitas ini dimulai dengan *employee* membuka aplikasi lalu *select project* pada aplikasi. Kemudian *employee* melakukan *start tracking* lalu pada aplikasi akan melakukan *capture application running, screen shoot* dan *capture screen monitor duration.* Selanjutnya aplikasi akan menyimpan data-data tersebut kedalam *database.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.9 *Activity Diagram Update Task Status*

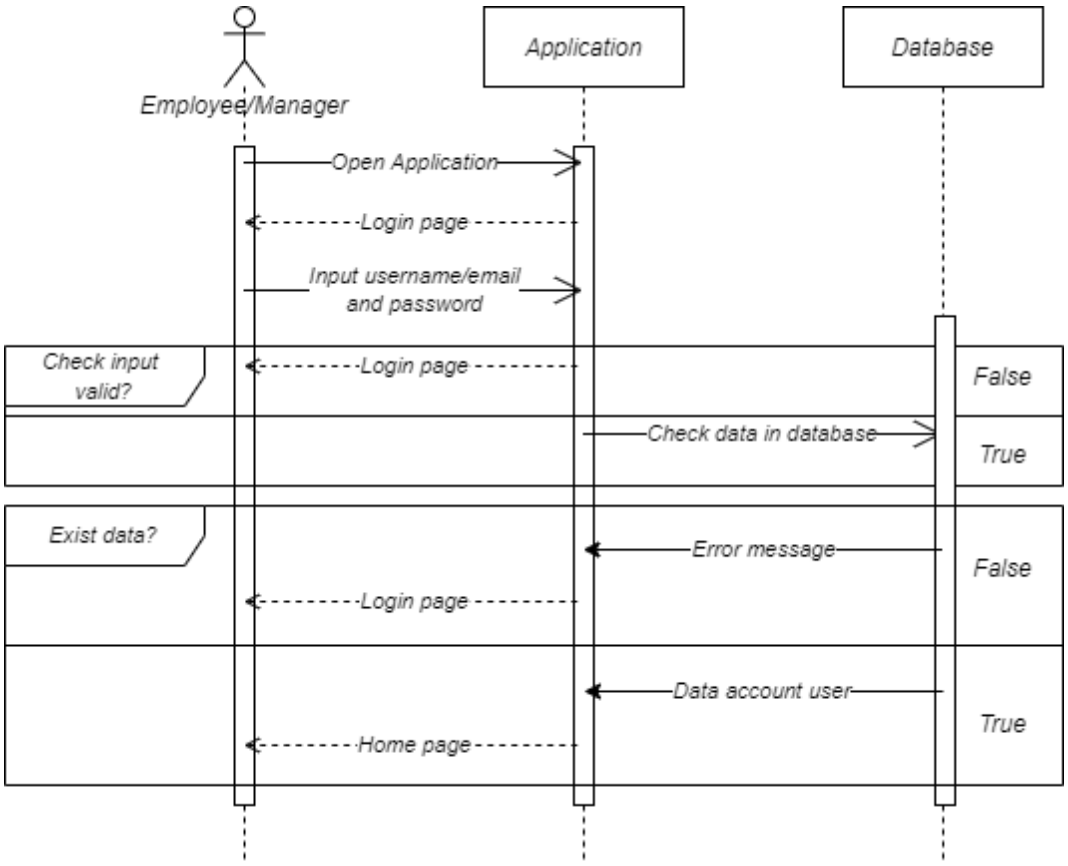
Pada gambar 4.9 digambarkan sebuah aktivitas *update task status* yang dilakukan oleh *employee*. Aktivitas pertama yang dilakukan yaitu membuka aplikasi lalu *select task* pada aplikasi. Kemudian *employee* melakukan konfirmasi *update status task* lalu pada aplikasi akan melakukan *check status task from database.* Jika *task status* pada database "*finish*", *task status* akan diubah menjadi "*on progress*". Begitu sebaliknya jika *task status* pada database "*on progress*", *task status* akan diubah menjadi "*finish*".

3. *Sequence diagram*

Pada *sequence diagram* ini memiliki dua objek yaitu *Application* dan *Database* beserta aktor yang menjalankan setiap proses. Desain *sequence diagram* didapatkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

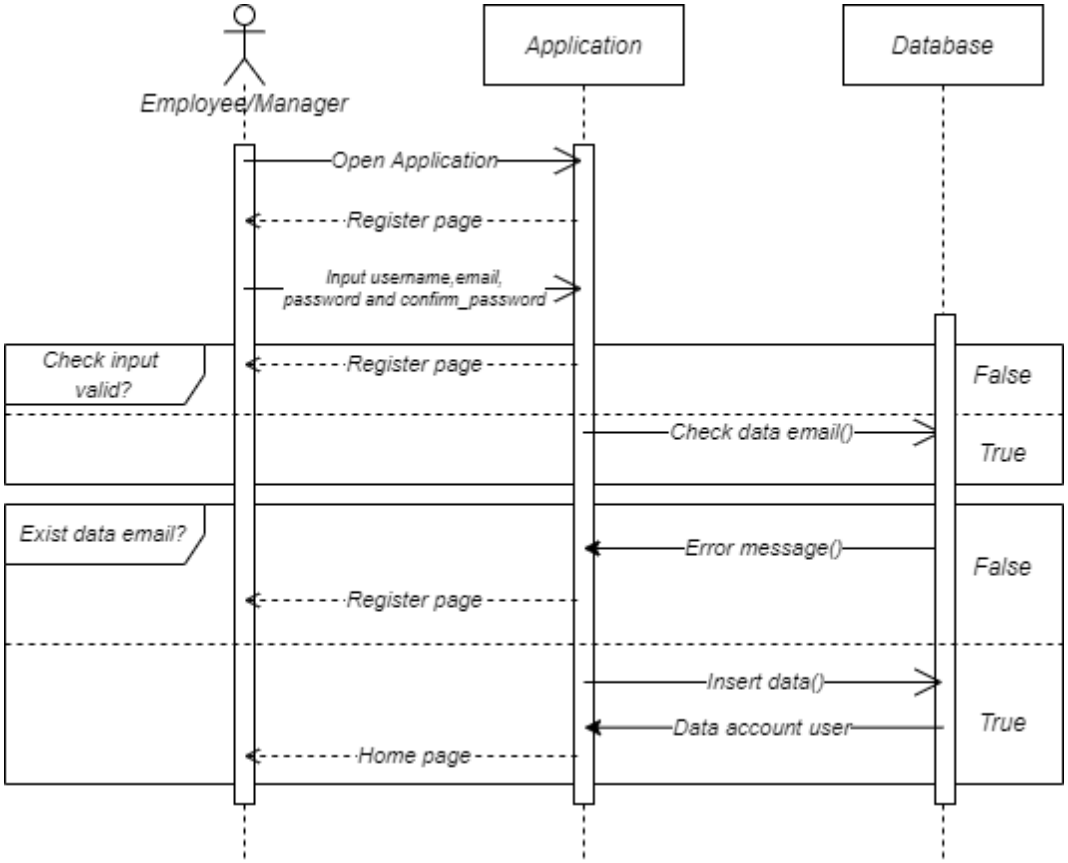
berdasarkan desain *activity diagram* yang sudah dibuta. *Sequence diagram* pada aplikasi ini terdiri dari *sequence diagram login, register, management project, management task, join project, tracking* dan *update status task.* Pada proses *login* dan *register,* pengguna dengan *role manager* dan *role employee* melakukan aktivitas yang sama disetiap prosesnya.



Gambar 4.10 *Sequence Diagram login*

Pada gambar 4.10 digambarkan sebuah proses *login* yang dilakukan oleh *employee* atau *manager*. Proses pertama yang dilakukan yaitu membuka aplikasi kemudian dari aplikasi akan menampilkan halaman *login.* Selanjutnya pengguna memasukan *username/email* dan *password,* lalu aplikasi akan melakukan *check inputan,* jika *valid* aplikasi akan *check data* pada database jika *invalid* aplikasi akan mengembalikan ke halaman *login.* Pada proses *check data in database,* jika datanya ada aplikasi akan menampilkan halaman *home,* jika datanya tidak ada aplikasi akan memunculkan pesan *error* dan mengembalikan pada halaman *login.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

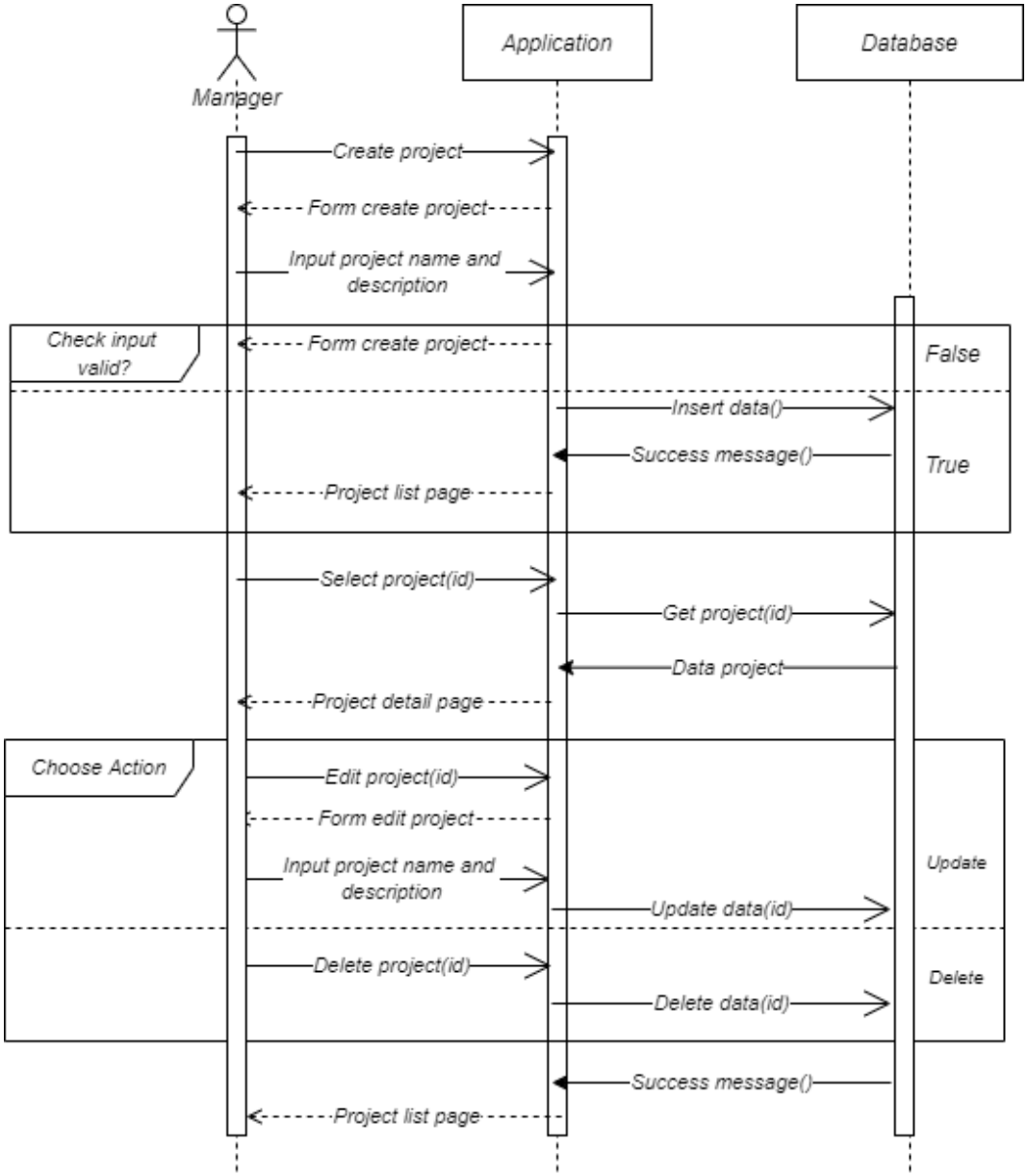


Gambar 4.11 *Sequence Diagram Register*

Pada gambar 4.11 digambarkan sebuah proses *register* yang dilakukan oleh *employee* atau *manager*. Proses pertama yang dilakukan yaitu membuka aplikasi kemudian dari aplikasi akan menampilkan halaman *register.* Selanjutnya pengguna memasukan *username, email, password* dan *confirm password,* lalu aplikasi akan melakukan *check inputan,* jika *valid* aplikasi akan *check data email* pada database jika *invalid* aplikasi akan mengembalikan he halaman *register.* Pada proses *check data email in database,* jika datanya tidak ada aplikasi akan menambahkan data lalu menampilkan halaman *home,* jika datanya ada aplikasi akan memunculkan pesan *error* dan mengembalikan pada halaman *register.*

Pada gambar 4.12 digambarkan sebuah proses *management project* yang dilakukan oleh *manager*. Proses pertama yang dilakukan yaitu memilih menu *management project* kemudian dari aplikasi akan menampilkan dua opsi yaitu *create project* dan *project list.* Pada aktivitas *create project, manager* akan memasukan nama proyek dan deskripsi pada *form* lalu aplikasi akan menyimpan data tersebut dalam database.

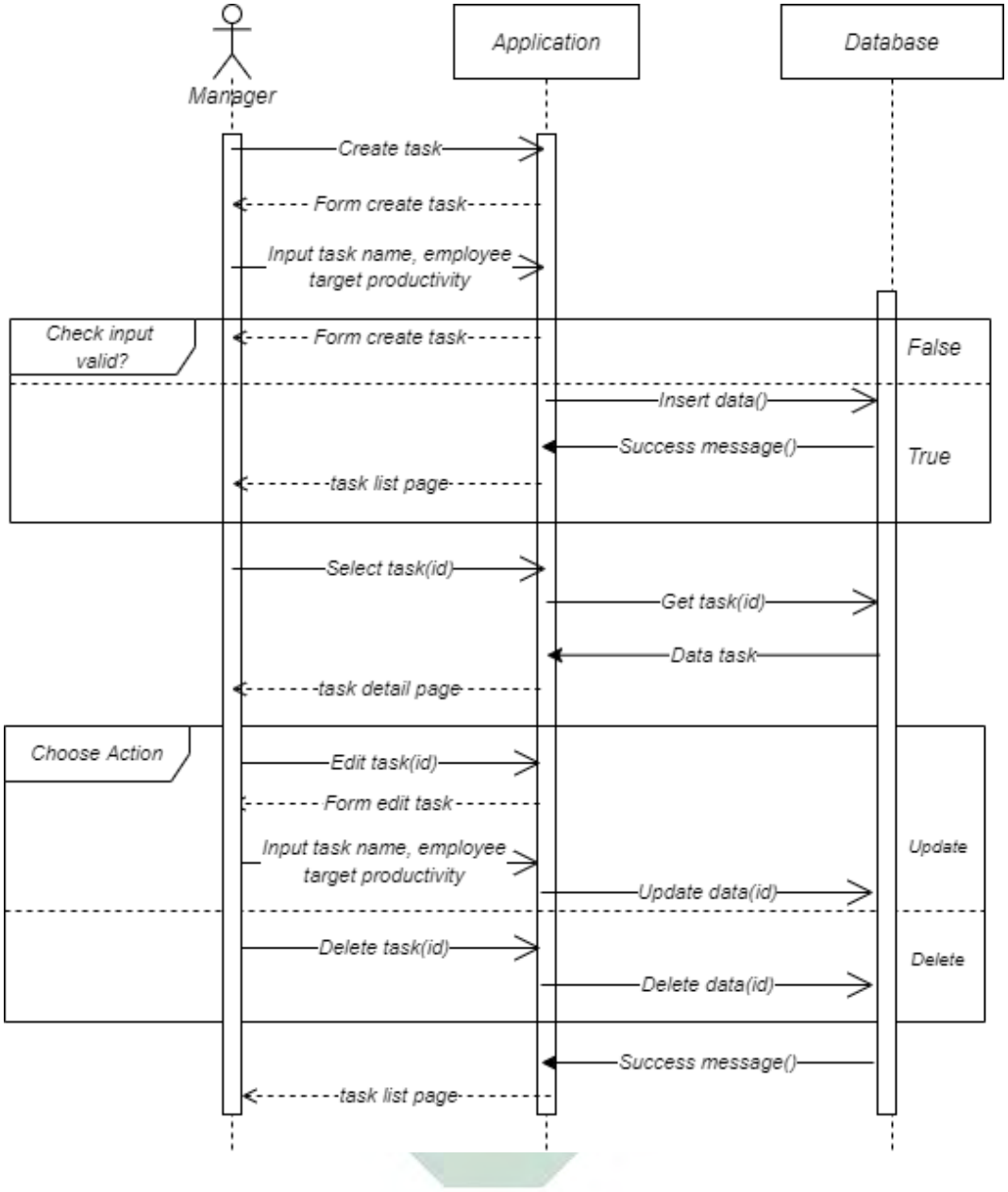
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.12 *Sequence Diagram Management Project*

Pada *project list, manager* akan memilih proyek lalu dari proyek tersebut memiliki dua opsi lagi yaitu *update project* atau *delete project.* Pada *update project, manager* akan mengisi *form* lalu aplikasi akan merubah data sesuai masukan dari *manager.* Kemudian pada proses *delete project, manager* akan mengkonfirmasi terlebih dahulu setelah itu aplikasi akan menghapus data yang ada dalam *database.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

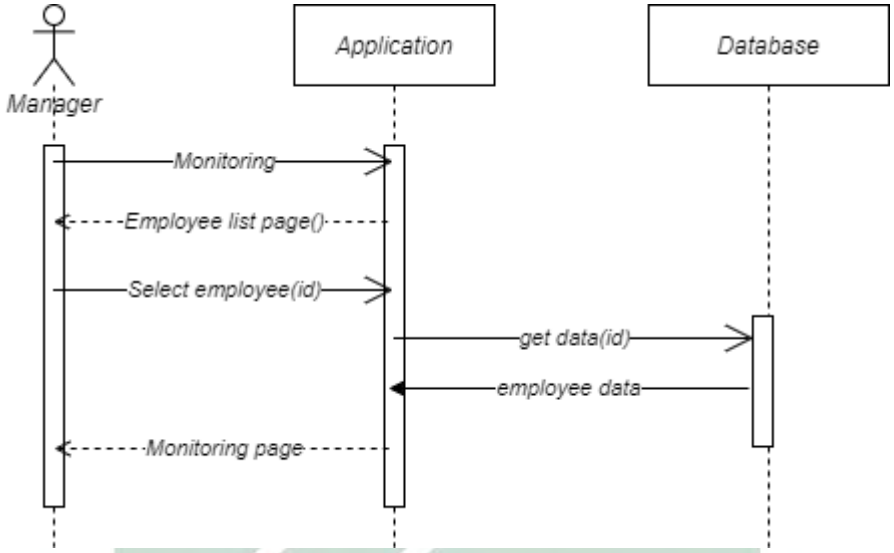


Gambar 4.13 *Sequence Diagram Management Task*

Pada gambar 4.13 digambarkan sebuah proses *management task* yang dilakukan oleh *manager*. Pada proses ini semua aktivitas proses hampir sama dengan *management project.* Perbedaan pada manament *task* yaitu pada bagian *form* memiliki atribut nama *task, employee* pada *task* dan target produktivitas karyawan.

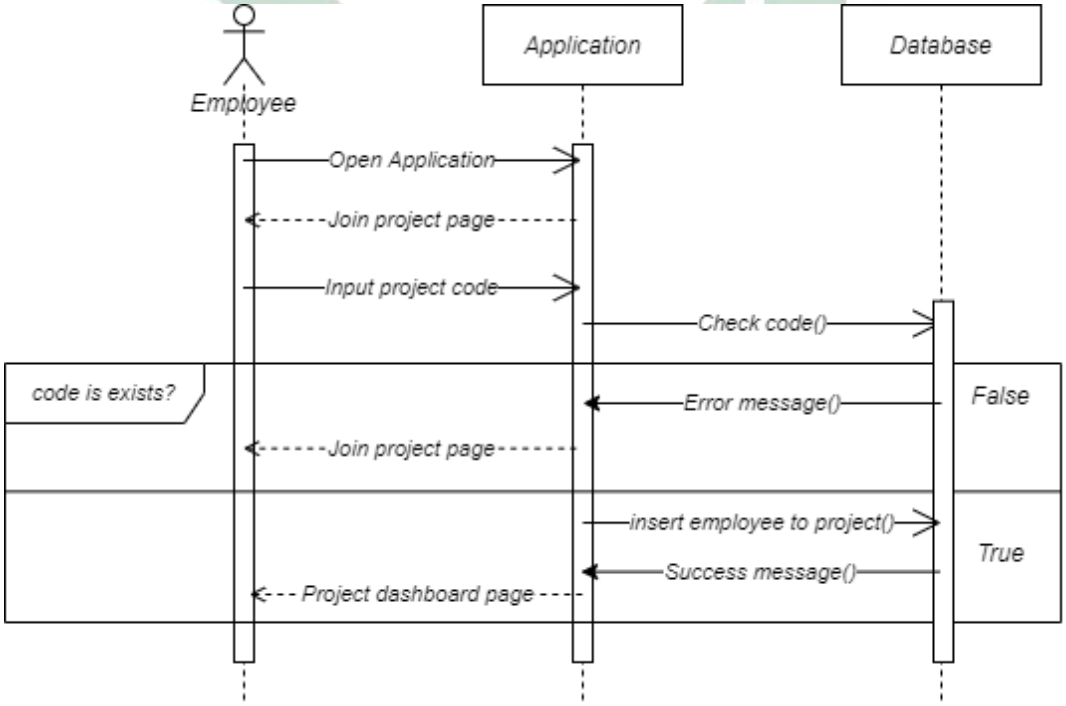
Pada proses *get task* atau *get project* memiliki parameter *id* untuk mendefinisikan data *task* atau data *project* tertentu. Begitu juga pada proses *edit task, edit project, delete task, delete project, update data* dan *delete data.*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.14 *Sequence Monitoring Employee*

Pada gambar 4.14 digambarkan sebuah proses *monitoring employee* yang dilakukan oleh *manager*. Proses pertama yang dilakukan yaitu memilih menu *monitoring* lalu pada aplikasi akan menampilkan *employee list.* Selanjutnya *manager* melakukan *select project* lalu pada aplikasi akan menampilkan detail statisistik *task, application open, screen shoot* dan *productivity employee.*

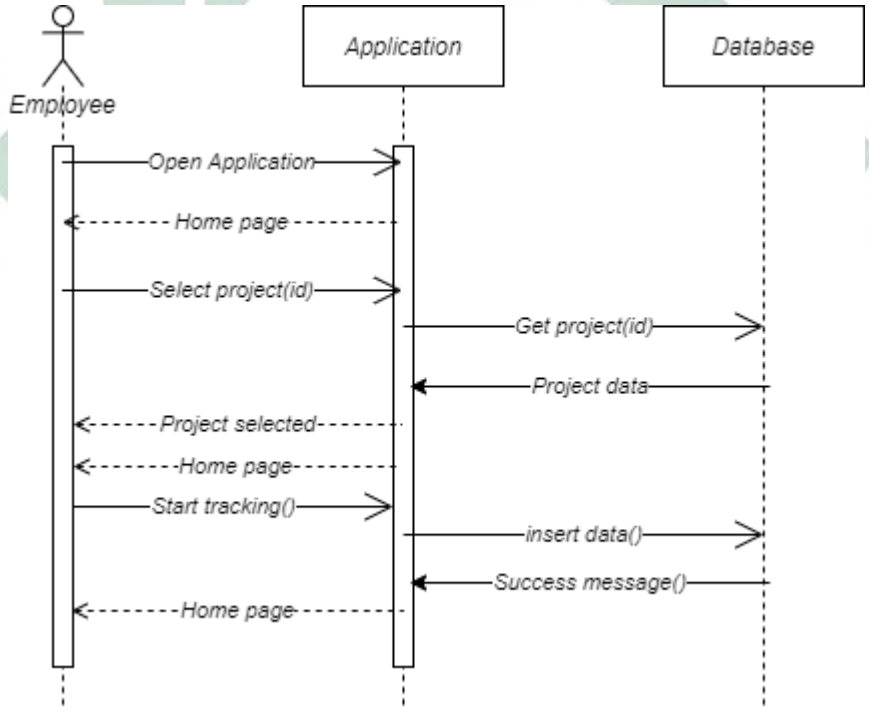


Gambar 4.15 *Sequence Diagram Join Project*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.15 digambarkan sebuah proses *join project* yang dilakukan oleh *employee*. Proses pertama yang dilakukan yaitu membuka aplikasi lalu pada aplikasi akan menampilkan halaman *join project.* Kemudian *employee* memasukan kode proyek lalu aplikasi akan melakukan *check code from databse,* jika kodenya sesuai, aplikasi akan melakukan *insert account employee to project*, jika tidak sesuai, aplikasi akan memunculkan pesan kesalahan dan mengembalikan pada halaman *join project.*

Pada proses *join project* yang dilakukan oleh *employee* memiliki proses yang dilakukan diluar aplikasi yaitu komunikasi antara *manager* dan *employee.* Proses komunikasi ini guna untuk memberitahukan kode proyek kepada *employee* oleh *manager project.* Dari situ *employee* dapat memasukan kode pada *form join project.*



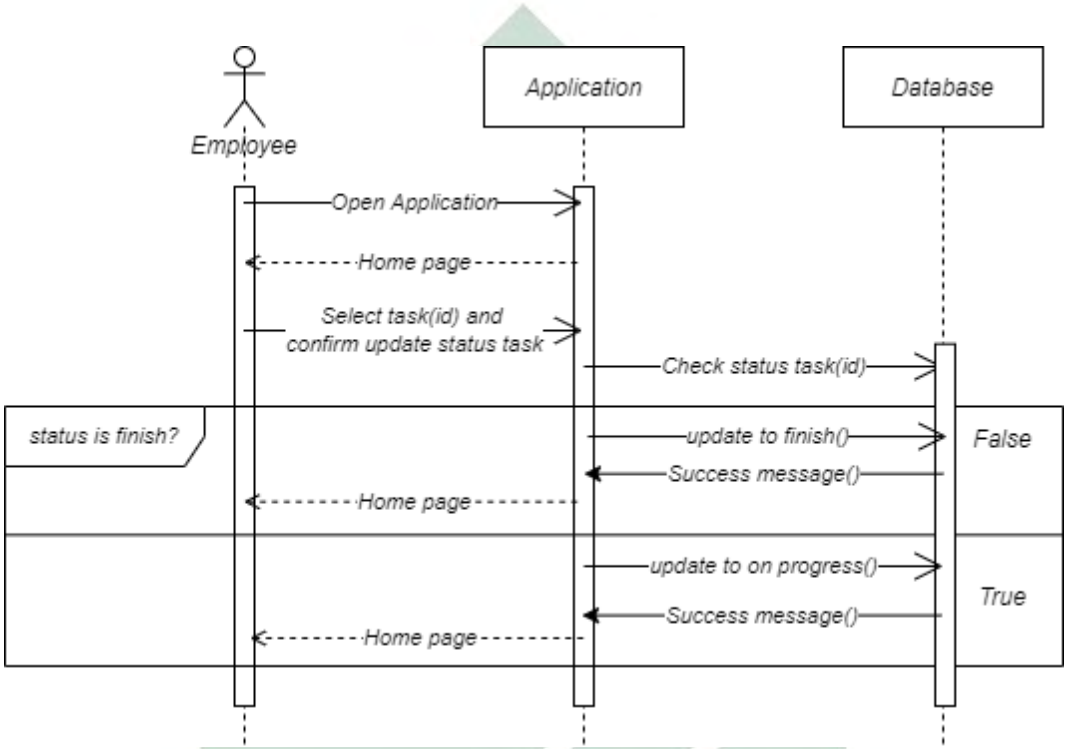
Gambar 4.16 *Sequence Diagram Tracking*

Pada gambar 4.16 digambarkan sebuah proses *tracking* yang dilakukan oleh *employee*. Pada proses ini dimulai dengan *employee* membuka aplikasi lalu *select project* pada aplikasi. Kemudian *employee* melakukan *start tracking* lalu pada aplikasi akan melakukan *capture application running, screen shoot* dan *capture*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

screen monitor duration. Selanjutnya aplikasi akan menyimpan data-data tersebut kedalam *database.*

Pada proses *start tracking, employee* akan melakukakan pekerjaanya pada komputer yang sudah terinstall aplikasi ini. Kemudian pada saat *tracking* berjalan, aplikasi akan melakukan *capture application running, capture monitor duration* dan *screen shoot* otomatis lalu data-data tersebut akan masuk pada *database.*



Gambar 4.17 *Sequence Diagram Update Status Task*

Pada gambar 4.17 digambarkan sebuah proses *update task status* yang dilakukan oleh *employee*. Proses pertama yang dilakukan yaitu membuka aplikasi lalu *select task* pada aplikasi. Kemudian *employee* melakukan konfirmasi *update status task* lalu pada aplikasi akan melakukan *check status task from database.* Jika *task status* pada database "*finish*", *task status* akan diubah menjadi "*on progress*". Begitu sebaliknya jika *task status* pada database "*on progress*", *task status* akan diubah menjadi "*finish*".

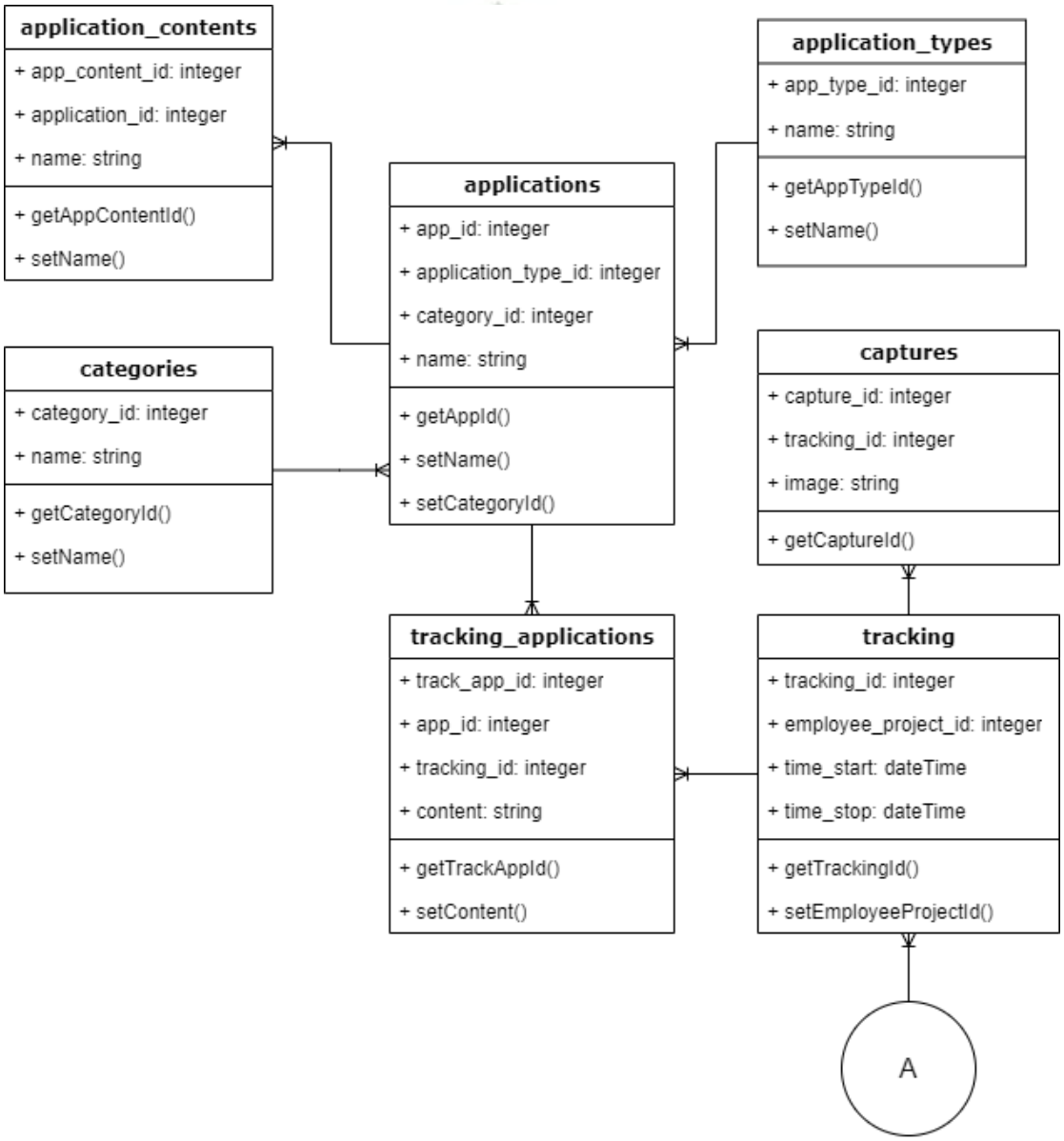
4. *Class diagram*

Class diagram ini di desain berdasarkan hasil analisis kebutuhan aplikasi pada tabel 4.1 dan desain *sequence diagram.* Dari analisis tersebut didapatkan data-data

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

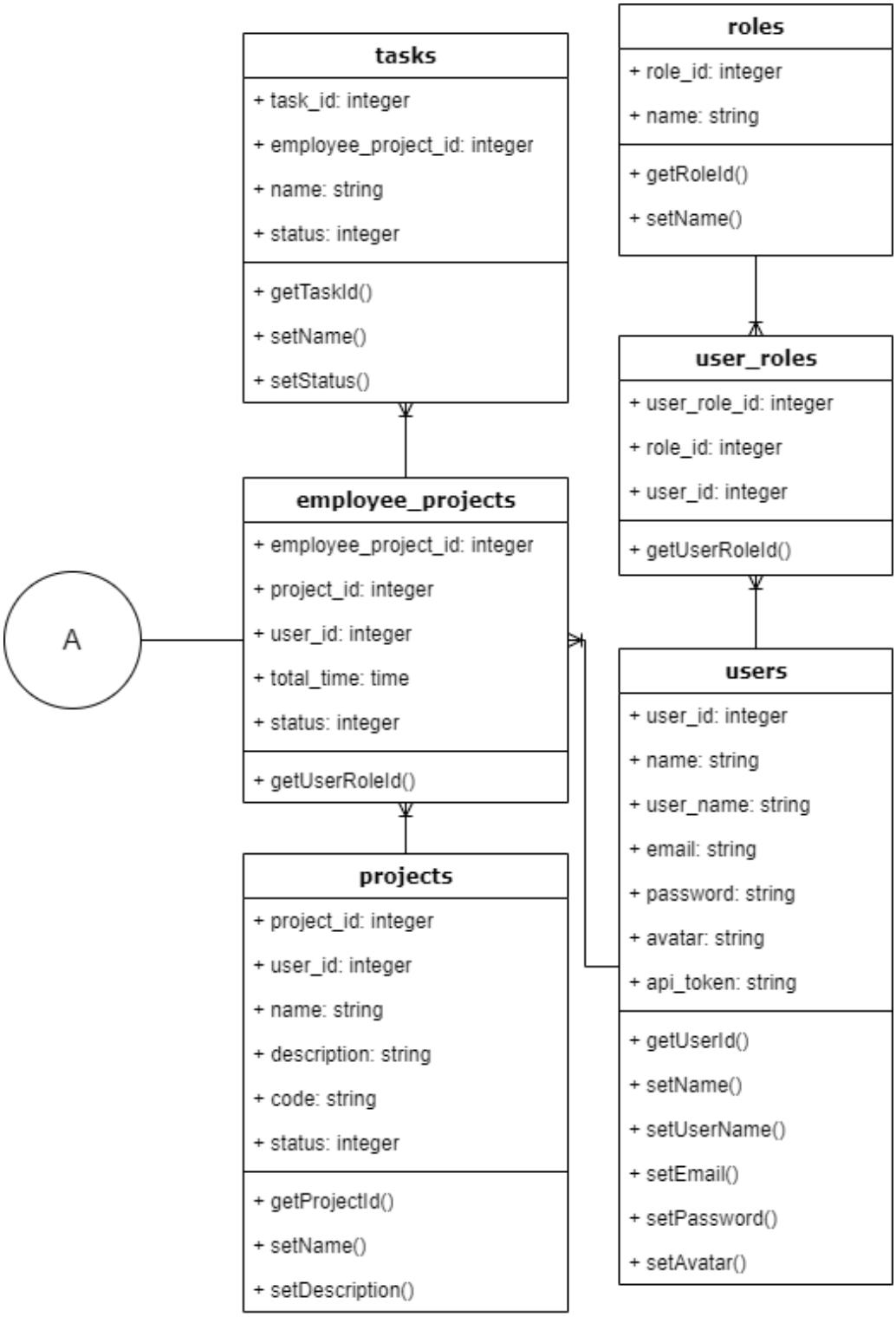
utama yang dibutuhkan oleh aplikasi beserta fungsi-fungsi dari setiap aktivitas proses yang dikerjakan.

Pada *class diagaram* memiliki sebanyak 12 kelas yaitu *application_contents, application_types, applications, categories, captures, tracking_applications, tracking, tasks, employee_projects, projects, users, user_roles,* dan *roles.*



Gambar 4.18 *Class Diagaram* Penelitian ini

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.19 Lanjutan *Class Diagaram* Penelitian ini

Pada gambar 4.18 digambarkan memiliki simbol koneksi dengan gambar 4.19 dimana *class tracking* berelasi dengan kelas *employee_project*. Kemudian dari setiap kelas memiliki atribut "*id"* dengan tipe data *integer* untuk dapat berelasi

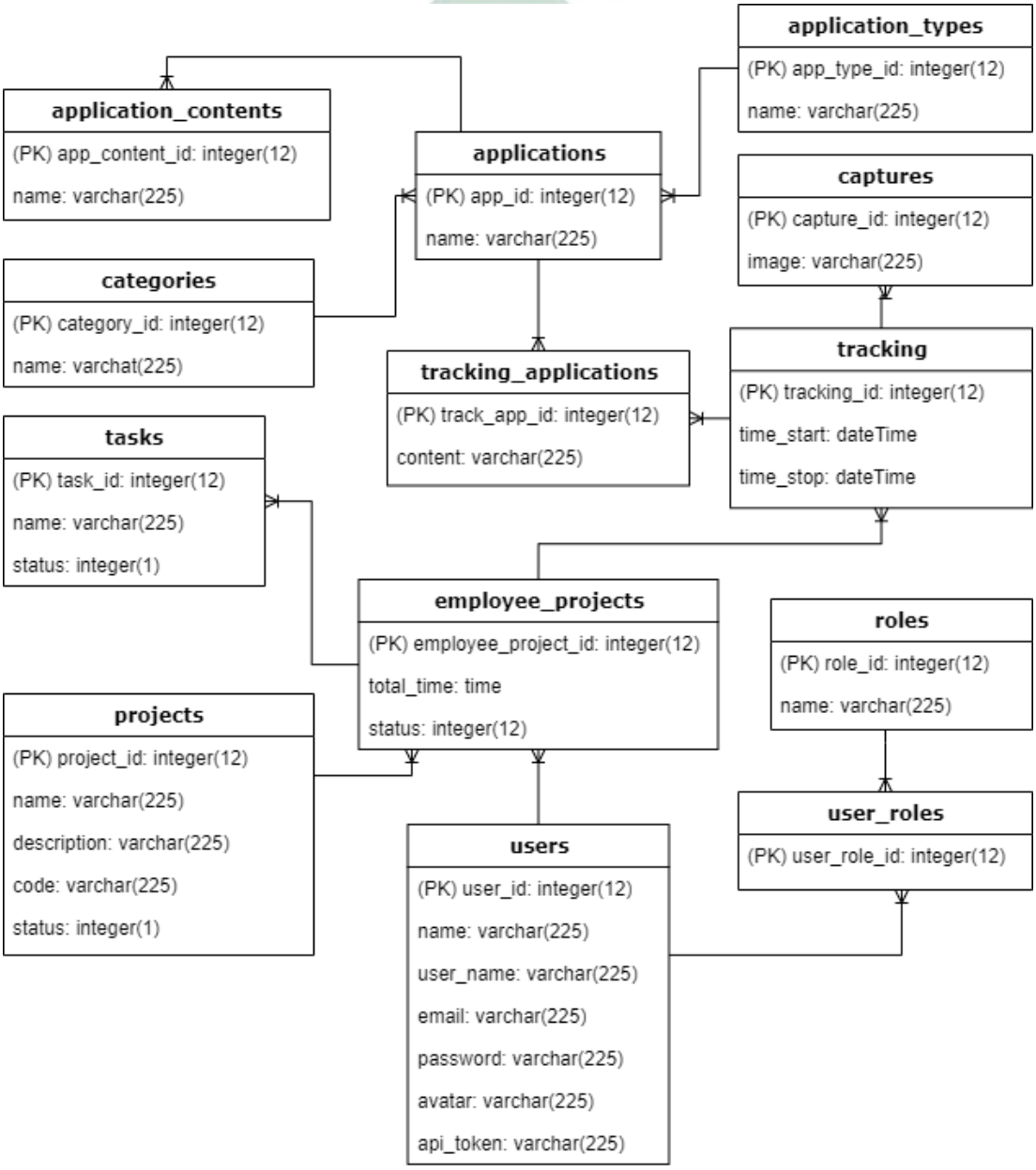
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan kelas yang lain. Pada setiap kelas juga memilik fungsi "*getId*" yang digunakan untuk mengambil data sesuai dengan *id* data yang diinginkan. Kemudian fungsi *"set"* digunakan untuk merubah nilai atribut terkait.

4.2.2 Desain *Database*

Desain *database* ini dirancang untuk membangun aplikasi yang terdiri dari desain *conceptual data model* dan *physical data model.*

1. *Conceptual diagram* (CDM)



Gambar 4.20 CDM Penelitian ini

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.20 dapat disimpulkan bahwa desain CDM ini memiliki tiga belas tabel yaitu *users, user_roles, roles, project, employee_project, application, application_types, application_content, tracking_applications categories, productivities, trackings* dan *captures.* Dari setiap tabel tersebut memiliki atribut, tipe data, *Primary Key* (PK) dan relasi.

Relasi antar tabel ini dinamakan dengan kardinalitas yang menyatakan suatu nilai hubungan relasi. Detail tabel-tabel yang memiliki relasi dapat dilihat pada tabel 4.5.

Tabel 4.5 Relasi Antar Tabel

From table	*To table*	*Cardinality*
Application_contents	*Applications*	*Many to one*
Applications	*Application_type*	*Many to one*
	categories	*Many to one*
Tracking_applications	*Application*	*Many to one*
	Tracking	*Many to one*
Tracking	*Employee_projects*	*Many to one*
Captures	*Tracking*	*Many to one*
Tasks	*Employee_projects*	*Many to one*
Employee_projects	*Project*	*Many to one*
	User	*Many to one*
Project	*User*	*Many to one*
User_roles	*Roles*	*Many to one*
	user	*Many to one*

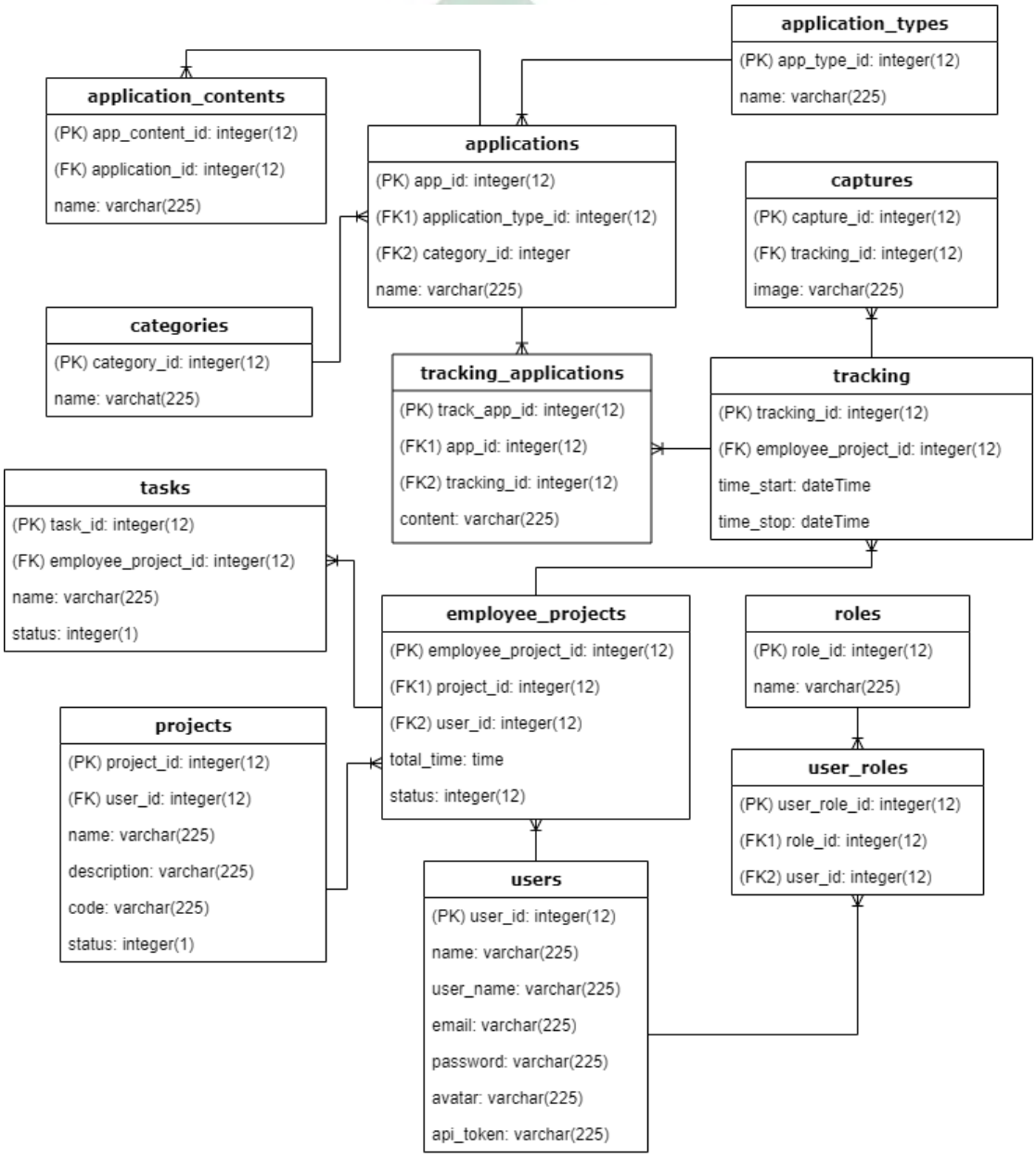
Dari tabel 4.5 dijelaskan bahwa relasi antar tabel dari tabel *application_contents* ke tabel *applications* memiliki hubungan *many to one* atau dapat dibalik dari tabel *applications* ke tabel *applications* memiliki hubungan *one to many* yang artinya satu data dari tabel *applications* dapat memiliki banyak data

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dari tabel *application_contents.* Pada tabel yang lain dengan kardinalitas sama, maka juga memiliki artian yang sama seperti relasi antara tabel *application_contents* dan *applications.*

2. *Physical data model* (PDM)

PDM ini diperoleh dari CDM yang dibuat dari tahap sebelumnya. Pada desain PDM ini sudah memiliki atribut *Foreign Key* (FK) pada setiap tabel yang memiliki relasi.



Gambar 4.21 PDM Penelitian ini

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.21 setiap tabel yang berelasi memiliki *Foreign Key* (FK) pada atribut tabel tersebut. Pada bagian inilah yang membedakan antara desain CDM dan PDM. Selain itu juga, desain PDM ini dapat di *export* menjadi *Structured Query Language* (SQL).

4.3 Implementasi

Pada tahap implementasi melakukan eksekusi dari tahapan analisis dan desain pada aplikasi yang dikembangkan hingga menghasilkan *prototype* aplikasi sesuai kebutuhan sistem. Implementasi dalam pengembangan aplikasi ini meliputi implementasi perangkat lunak dan perangkat keras, implementasi metode SVM dalam aplikasi serta implementasi *user interface.*

4.3.1 Implementasi SVM

Pada implementasi metode SVM dilakukan tahap *preprocessing data* dan pelatihan model dari dataset yang sudah didapatkan. Pada tahap *preprocessing data* dilakukan dengan metode *data reduction* dan pelatihan model akan dilakukan proses *training* dan *testing.*

1. *Preprocessing data*

Preprocessing data dilakukan dengan metode *data reduction* dengan menyaring variable-variabel yang diperlukan dalam pelatihan model. *Data reduction* ini digunakan untuk meningkatkan efektivitas *machine learning* dalam klasifikasi kinerja karyawan. Pada proses ini dilakukan penghapusan *column* yang bukan termasuk *independent variable* dalam dataset tersebut supaya tidak ada variable yang menyebabkan ke ambiguan dalam klasifikasi.

Setelah dilakukan tahapan ini, dataset yang awalnya memiliki 15 *column* variabel yang terdiri dari *date, quarter, department, day, team, targeted_productivity, smv, wip, over_time, incentive, idle_time, idle_men, no_of_style_change, no_of_workers,* dan *actual_productivity* menjadi 4 *column* yaitu *target_productivity, smv, wip,* dan *actual_productivity* dengan hasil dataset seperti pada tabel 4.6

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.6 Dataset setelah tahap *preprocessing data*

data ke- (n)	*Target Productivity* ($x1$)	*Standart minute value* ($x2$)	*Work in progress* ($x3$)	*Over Tme (x4)*	*No of Worker (x5)*	*Actual Productivity* (y)
1	0.8	26.16	1108	7080	59	0.940725
2	0.75	3.94	0	960	8	0.8865
3	0.8	11.41	968	3660	30.5	0.80057
4	0.8	11.41	968	3660	30.5	0.80057
5	0.8	25.9	1170	1920	56	0.800382
6	0.8	25.9	984	6720	56	0.800125
….						
1197	0.7	2.9	0	720	6	0.394722222

Pada *column* variabel *date, quarter, department,* dan *day* dihapus karena memiliki nilai *character* yang tidak bisa digunakan dalam pelatihan model. Pada *column* variabel *team, incentive, idle_team,* dan *idle_men* dihapus dikarenakan pada penelitian ini tidak menggunakan variabel tersebut dalam proses penentuan kinerja karyawan.

2. Pelatihan model

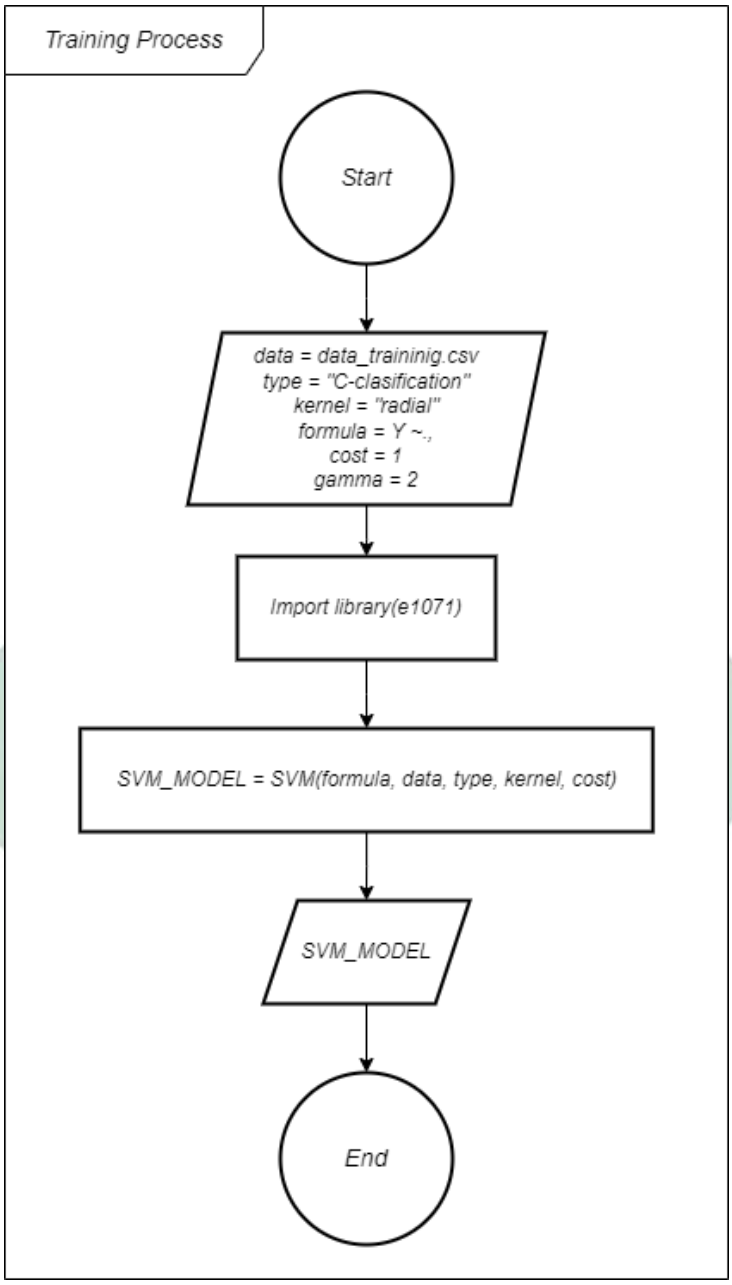
Pelatihan model digunakan untuk melakukan klasifikasi dengan menggunakan data hasil kerja karyawan yang terdiri dari empat variabel yaitu *target_productivity* ($x1$), *smv* ($x2$), *wip* ($x3$) dan *actual_productivity* (y) kemudian dilakukan proses *training* dan *testing*. Proses *training* dan *testing* menggunakan dataset pada tabel 4.6 dengan membaginya 90: 10, dengan 90% data untuk *training* (*data*_training.csv) dan 10% data untuk *testing (data_testing.csv).* Perbandingan pembagian ini dan inisialisasi variabel adalah paling optimal dengan performa terbaik yang dibuktikan dengan pengujian.

a. Proses *Training*

Proses *training* bertujuan untuk mendapatkan sebuah model yang digunakan untuk proses *testing* dengan menggunakan data *testing*. Sebelum melakukan proses *training* perlu menyiapkan aplikasi *R Studio, library e1071,* dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

training dataset. Proses *training* dapat dijelaskan pada gambar 4.9 *flowchart training process.*



Gambar 4.22 *flowchart training process*

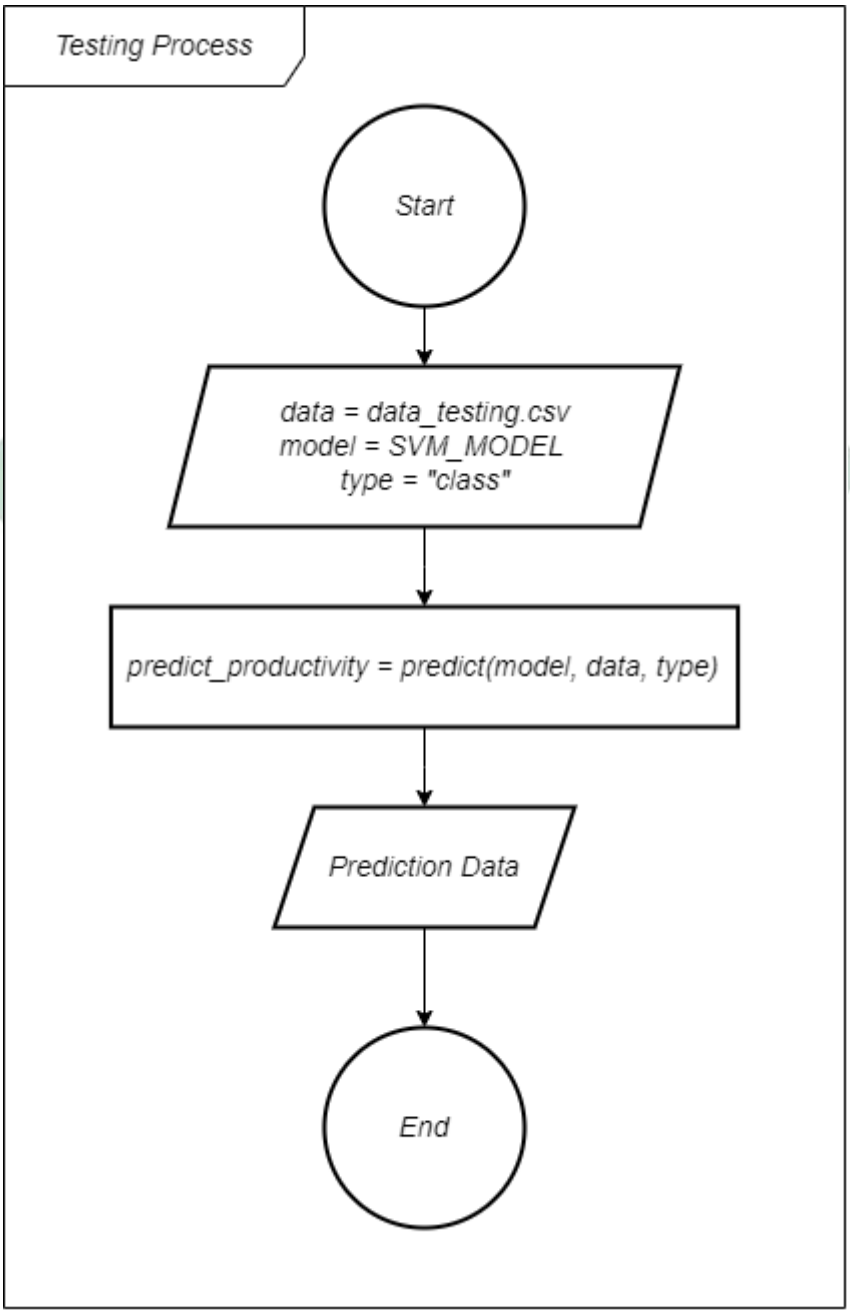
Pada gambar 4.22 proses dimulai dari inisialisasi variabel inputan yaitu *data, type, kernel, formula* dan *cost. Data* adalah dokumen *training_dataset.csv* yang didapat 90% data dari dataset tabel 4.6 dan *type* adalah "*C-clasification*" untuk mengidentifikasi bahwa pelatihan model menggunakan klasifikasi dengan *formula "y"* berarti variabel *y* yaitu *actual_productivity.* Setelah itu *import library e1071, library* ini digunakan untuk menghasilkan fungsi untuk membuat model SVM.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Selanjutnya masukan data variabel inputan kedalam fungsi SVM dan menghasilkan sebuah keluaran variabel *SVM_MODEL.*

b. Proses *testing*

Pada proses *testing* ini akan memprediksi dari *data_testing.csv* yang memiliki jumlah 10% data dari dataset tabel 4.6. pada tahap ini akan diklasifikasi apakah data termasuk dalam kategori *productive* atau *unproductive.* Proses *testing* dapat dijelaskan pada gambar 4.23 *flowchart testing process.*



Gambar 4.23 *flowchart testing process*

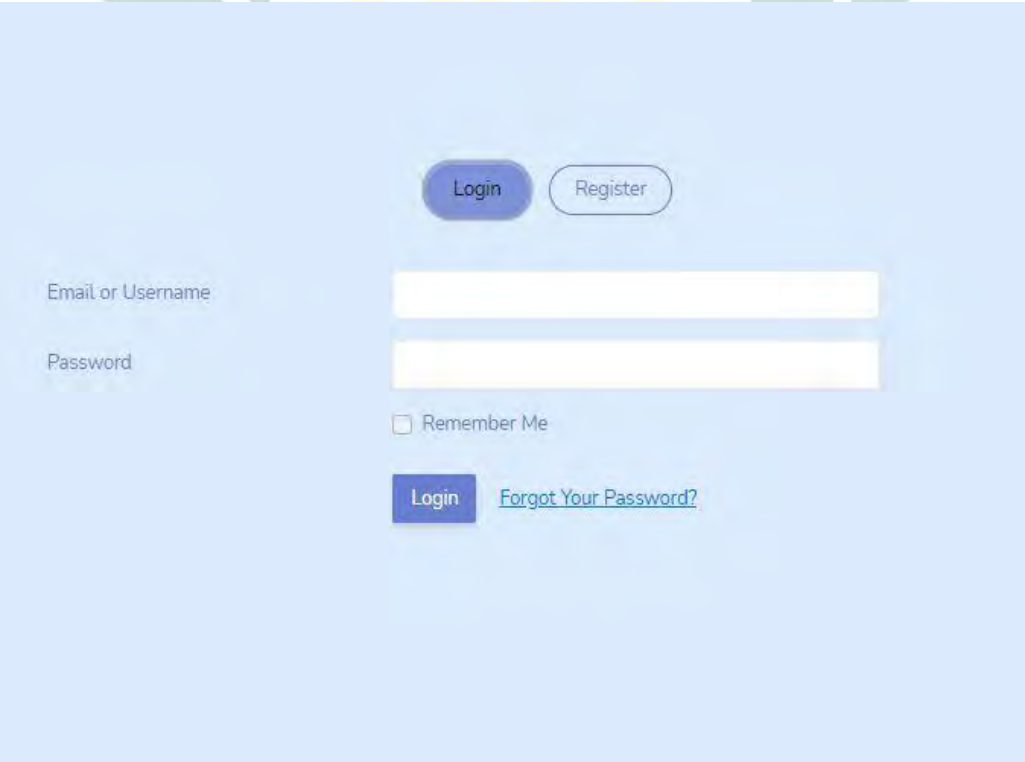
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.23 proses dimulai dari inisialisasi variabel inputan yaitu *data, model, target_productivity* dan *type. Data* adalah dokumen *testing_dataset.csv* yang didapat 30% data dari dataset tabel 4.6 dengan *type* prediksi "*class*" dan model menggunaakan variabel SVM_*MODEL* yang didapat dari tahap proses *traininig*. Selanjutnya masukan variabel inputan tersebut pada fungsi *predict* untuk memprediksi nilai prediksi prodtivitas dan menghasilkan keluaran variabel *predict_productivity*. Terakhir buat kondisi untuk klasifikasi dimana jika $predic_productivty < target_productivity$, maka masuk dalam kategori *unproductive,* jika sebaliknya maka masuk dalam kategori *productive.*

4.3.2 Implementasi *User Interface*

Pada tahap implementasi *user interface* melakukan eksekusi program dengan membuat tampilan dan fungsi-fungsi didalamnya. Tampilan tersebut meliputi *login & register web app, working board, management task, monitoring page, monitoring employee, login desktop app, home page* dan *time working.*

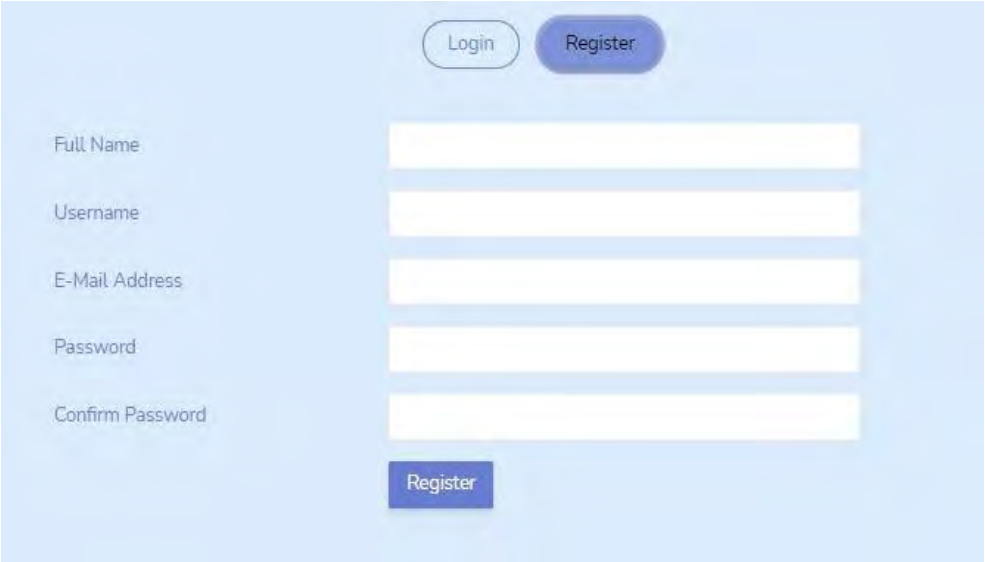
a. *Login & register web app*



Gambar 4.24 *Login Web App*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

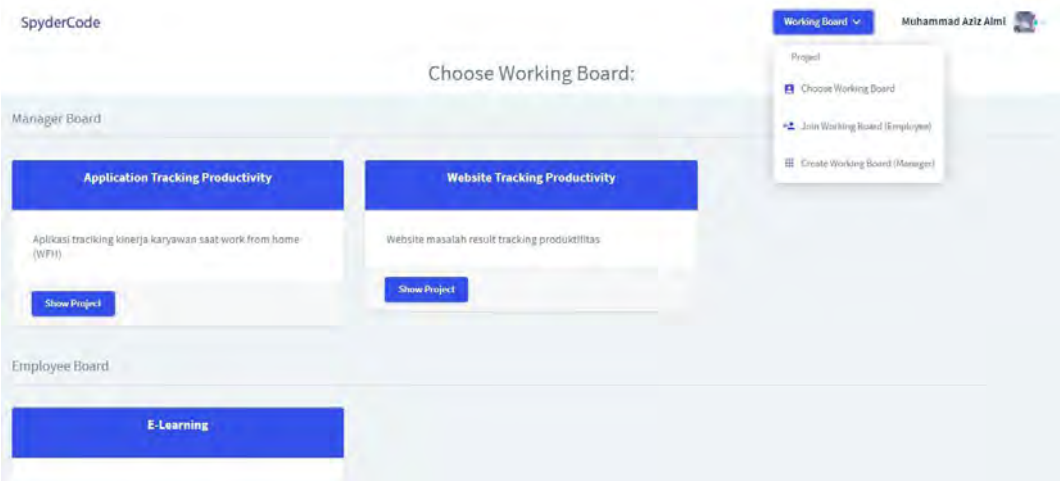
Pada gambar 4.24 adalah halaman *login user* untuk masuk ke dalam dashboard aplikasi. Pada halaman ini memiliki *form* pengisian *email* atau *username* dan *password* yang digunakan sebagai akun untuk masuk aplikasi. Kemudian ada *button register* untuk pindah ke halaman registrasi.



Gambar 4.25 *Register Web App*

Pada gambar 4.25 adalah halaman registrasi *user* dengan mengisi *form* yang berisi *full name, username, email, password* dan *confirm password.* Kemudian ada *button login* yang berfungsi untuk pindah ke halaman *login.*

b. *Working board*

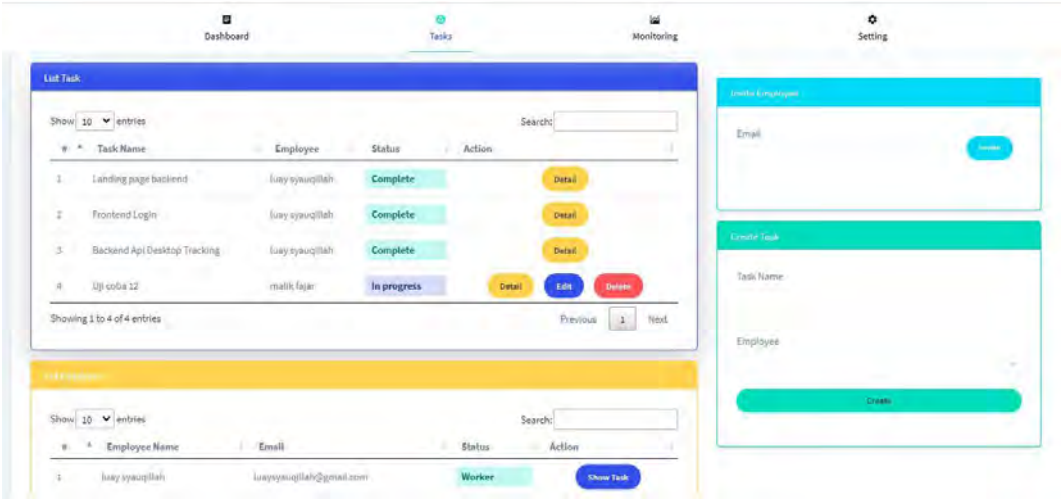


Gambar 4.26 *Working Board*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.26 adalah halaman *working board* yang berfungsi sebagai halaman utama aktivitas yang akan dikerjakan. Pada halaman ini ada menu untuk masuk dalam proyek sebagai *employee* atau sebagai *manager* dengan proyek-proyek tertentu. Kemudian dalam halaman ini juga ada navigasi untuk membuat proyek baru dan *join project as employee.*

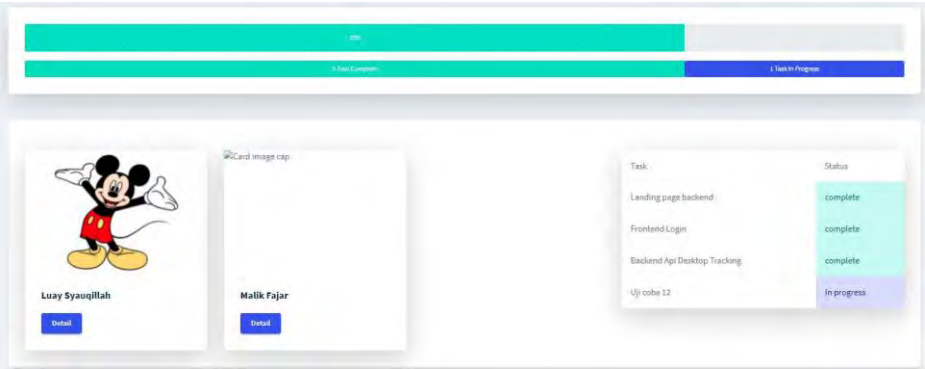
c. *Management task*



Gambar 4.27 *management task*

Pada gambar 4.27 adalah halaman *management task* dimana pada halaman ini hanya bisa diakses oleh *manager* dalam sebuah proyek. Halaman ini juga memiliki beberapa fungsi yang diantaranya yaitu menambahkan *employee* ke dalam proyek, membuat *task* pada *employee,* melihat *task list,* merubah atau menghapus *task* dan melihat *employee list.*

d. *Monitoring page*

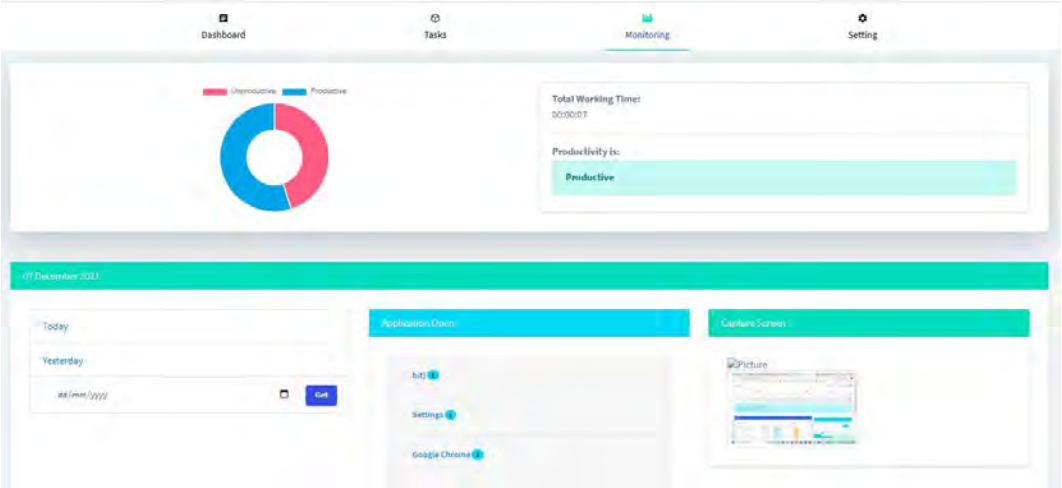


Gambar 4.28 *Monitoring page*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.28 adalah halaman utama untuk monitoring *task* yang ada pada sebuah proyek. Pada halaman ini menampilkan *task list* beserta status *task* tersebut, menampilkan *employee list* dan menampilkan statistik *task* yang sudah dan belum dikerjakan.

e. *Monitoring employee*



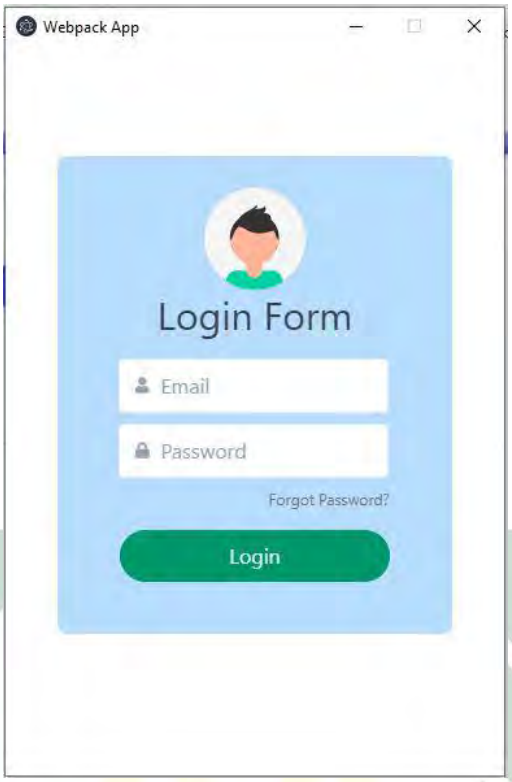
Gambar 4.29 *Monitoring Employee*

Pada gambar 4.29 adalah halaman untuk memonitoring karyawan oleh *manager* atau melihat detail pekerjaan oleh *employee*. Pada halaman ini menampilkan statistik produktivitas karyawan, total waktu karyawan selama bekerja, *application open list, screen shoot list* dan *task list* yang dikerjakan karyawan. Pada halaman ini juga memiliki navigasi tanggal untuk melihat hasil kerja pada tanggal tertentu.

f. *Login desktop app*

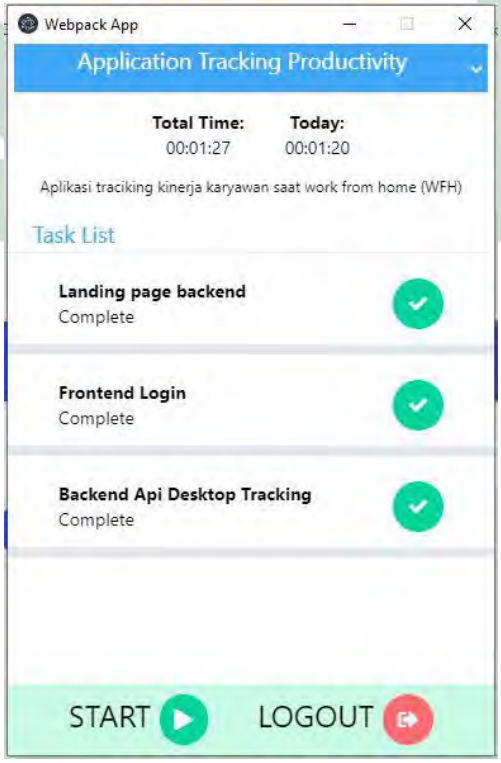
Pada tampilan *login desktop app* berbeda dengan gambar 4.24 dari segi fungsi dan tampilan. Pada halaman ini digunakan untuk *platform desktop* yang memiliki sistem operasi *windows.* Pada gambar 4.30 adalah halaman *login* untuk para *employee* yang akan melakukan pekerjaan menggunakan aplikasi ini. Pada halaman ini *login* menggunakan *email* dan *password* karyawan yang sudah melakukan regsitrasi pada gambar 4.25

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.30 *Login Desktop App*

g. *Home page*

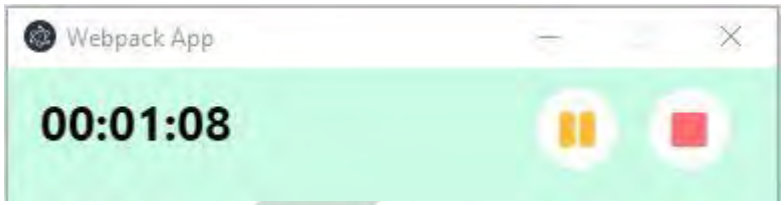


Gambar 4.31 *Home Page*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada gambar 4.31 adalah halaman utama aplikasi yang memiliki beberapa fungsi yaitu memilih proyek yang akan digunakan, mengerjakan *task,* mengubah status *task, start working* dan keluar aplikasi. Pada halama ini juga menampilkan detail proyek beserta lama waktu karyawan bekerja.

h. *Time working*



Gambar 4.32 *Time Working*

Pada gambar 4.32 adalah tampilan ketika *employee* melakukan *start working.* Pada halaman ini memiliki *timer* yang berjalan dimana saat waktu berjalan, aplikasi akan melakukan *capture application running* dan *screen shoot* secara otomatis dengan interval menit tertentu.

4.4 Pengujian

Pada tahap pengujian melakukan pengujian pada aplikasi yang dihasilkan pada tahap implementasi. Pengujian ini dilakukan untuk memastikan bahwa aplikasi berjalan sesuai dengan yang diinginkan. Pengujian dalam pengembangan aplikasi ini meliputi pengujian fungsioanal, pengujian RMSE dan pengujian performa.

4.4.1 Pengujian Fungsional

Pengujian fungsional berfokus pada pengujian dalam kegiatan-kegiatan yang dilakukan oleh *employee* dan *manager* dengan fitur-fitur yaitu *login user, registrasi user, start working* dan *result work.*

1. Pengujian fitur *login user*

Pada fitur ini akan diuji dengan tiga aksi yaitu memasukan email dan kata sandi yang sesuai database, memasukan email dan kata sandi kosongan dan memasukan email dan kata sandi yang salah. Hasil pengujian fitur ini dapat dilihat pada table 4.7

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Table 4.7 Pengujian fitur *login user*

No	*Action*	*Inputs*	*Expected Output*	*Actual Output*	*Test Result*
1	Masukan Email & Kata Sandi yang ada dalam database dan tekan tombol masuk	Email: almi@yahoo.com Password: ******	Masuk halaman home *page employee*	Masuk halaman home *page employee*	*Pass*
2	Masukan Email & Kata Sandi kosongan dan tekan tombol masuk	Email: Password:	Muncul pesan *error*	Muncul pesan *error*	*Pass*
3	Masukan Email & Kata Sandi yang tidak valid dan tekan tombol masuk	Email: aziz@yahoo.com Password: ******	Muncul pesan *error*	Muncul pesan *error*	*Pass*

2. Pengujian fitur *register user*

Pada fitur ini akan diuji dengan empat aksi yaitu mengosongkan nama, email, password, dan password konfirmasi; memasukan nama, email, password, dan password konfirmasi; memasukan password kurang dari enam karakter; dan memasukan email yang sudah terdaftar. Hasil pengujian fitur ini dapat dilihat pada table 4.8

Table 4.8 Pengujian fitur *register user*

No	*Action*	*Inputs*	*Expected Output*	*Actual Output*	*Test Result*
1	Masukan password kurang dari 6 karakter	Password: ****	Muncul pesan *error*	Muncul pesan *error*	*Pass*
2	Masukan email yang sudah terdaftar	Email: almi@yahoo.com	Muncul pesan *error*	Muncul pesan *error*	*Pass*
3	Mengosongkan Nama, Email, Password, dan Password konfirmasi dan tekan tombol masuk	Nama: Email: Password: Password Confirmation:	Muncul pesan *error*	Muncul pesan *error*	*Pass*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4	Masukan Nama, Email, Password, dan Password konfirmasi dan tekan tombol register	Nama: almi Email: almi@gmail.com Password: almi1234 Password Confirmation: almi1234	Masuk dalam halaman *home page*	Masuk dalam halaman *home page*	*Pass*

3. Pengujian fitur *start working*

Pada fitur ini akan diuji dengan tiga aksi yaitu menekan tombol "mulai", menangkap hasil *application running* saat waktu berjalan, menangkap layer dalam bentuk gambar saat waktu berjalan. Hasil pengujian fitur ini dapat dilihat pada table 4.9

Table 4.9 Pengujian fitur *start working*

No	*Action*	*Inputs*	*Expected Output*	*Actual Output*	*Test Result*
1	Menekan tombol "mulai"	Tekan tombol "mulai"	Muncul *timer* yang berjalan mulai dari 00:00:00	Muncul *timer* yang berjalan mulai dari 00:00:00	*Pass*
2	Menangkap hasil *application runnig*	*Capture interval: 1* (waktu berjalan tiap 1 menit akan menangkap hasil *application running*)	List *application running* masuk ke *database*	List *application running* masuk ke *database*	*Pass*
3	Menangkap hasil *screen shhot*	*Capture interval: 1* (waktu berjalan tiap 1 menit akan menangkap hasil *screen shoot*)	Hasil *screen shoot* masuk ke dalam *server* dan *database*	Hasil *screen shoot* masuk ke dalam *server* dan *database*	*Pass*

4. Pengujian fitur *result work*

Pada fitur ini akan diuji dengan tiga aksi yaitu menekan tombol "*show report*" dan menekan tombol "*show productivity percentage*". Hasil pengujian fitur ini dapat dilihat pada table 4.10

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Table 4.10 Pengujian fitur *result work*

No	*Action*	*Inputs*	*Expected Output*	*Actual Output*	*Test Result*
1	Menekan tombol "*show report*"	Tekan tombol "*show report*"	Muncul *report* dari karyawan	Muncul *report* dari karyawan	*Pass*
2	Menekan tombol "*show productivity percentage*"	Tekan tombol "*show productivity percentage*"	Muncul hasil perhitungan persentase produktivitas karyawan	Muncul hasil perhitungan persentase produktivitas karyawan	*Pass*

Dari hasil pengujian dari masing-masing fitur dengan total ada 12 aksi dapat disimpulkan bahwa pengujian fungsional tidak ditemukan *error* atau tidak sesuai harapan. Dengan begitu, tingkat keberhasilan pengujian fungsional ini mencapai 100%.

4.4.2 Pengujian RMSE

Pengujian *Root Mean Square Error* (RMSE) dilakukan dengan menguji dua model metode klasifikasi yaitu *Support Vector Machine* (SVM) dan *Support Vector Regression* (SVR). Pengujian RMSE ini dilakukan dengan aplikasi *R Studio* dengan data yang digunakan masing-masing model adalah sama. Pada pengujian ini menggunakan persamaan 2.2. yang mana perhitungan ada pada setiap skenario pengujian RMSE. Pada pengujian ini akan dilakukan tiga skenario dengan membagi proporsi rasio *data training* dan *data testing* antara lain skenario pertama $50:50$, skenario kedua $70:30$, dan skenario ketiga $90:10$.

a. Skenario pertama $50:50$

Pada skenario pertama dataset dibagi dengan proporsi 50% *data training* dan 50% *data testing.* Hasil dari skenario ini dapat dilihat pada tabel 4.9

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.11 Skenario 50: 50

Diketahui	Rumus	Nilai RMSE
$MSE = 0.2123746$	$RMSE = \sqrt{0.2123746}$	0.4608412

Pada tabel 4.11 dijelaskan bahwa nilai RMSE adalah 0.4608412 pada skenario 50: 50 proporsi rasio data *training* dan data *testing* dengan nilai MSE sebesar 0.2123746.

b. Skenario kedua 70: 30

Pada skenario pertama dataset dibagi dengan proporsi 70% *data training* dan 30% *data testing*. Hasil dari skenario ini dapat dilihat pada tabel 4.10

Tabel 4.12 Skenario 70: 30

Diketahui	Rumus	Nilai RMSE
$MSE = 0.2055556$	$RMSE = \sqrt{0.2055556}$	0.4533824

Pada tabel 4.12 dijelaskan bahwa nilai RMSE adalah 0.4533824 pada skenario 70: 30 proporsi rasio data *training* dan data *testing* dengan nilai MSE sebesar 0.2055556.

c. Skenario ketiga 90: 10

Pada skenario pertama dataset dibagi dengan proporsi 90% *data training* dan 10% *data testing*. Hasil dari skenario ini dapat dilihat pada tabel 4.13

Tabel 4.13 Skenario 90: 10

Diketahui	Rumus	Nilai RMSE
$MSE = 0.1848739$	$RMSE = \sqrt{0.1848739}$	0.4299697

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada tabel 4.13 dijelaskan bahwa nilai RMSE adalah 0.4299697pada skenario $90:10$ proporsi rasio data *training* dan data *testing* dengan nilai MSE sebesar 0.1848739.

Hasil dari pengujian RMSE menunjukkan bahwa pelatihan model menggunakan metode SVM menghasilkan nilai RMSE terbaik pada skenario ketiga karena pada perbandingan masing-masing skenario nilai RMSE terkecil menunjukan pada skenario ketiga dengan $90:10$ rasio proporsi data *training* dan data *testing* yang menghasilkan nilai RMSE sebesar 0.4299697.

4.4.3 Pengujian Performa

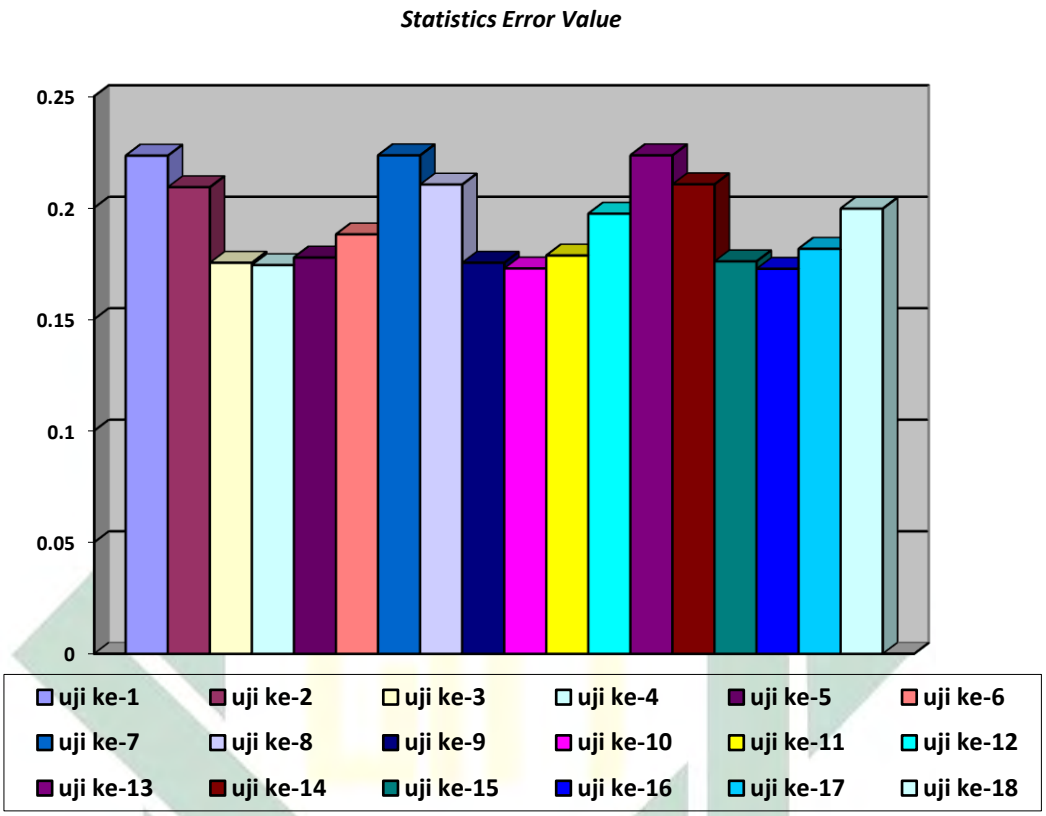
Pengujian performa dilakukan untuk mengetahui parameter terbaik untuk diterapkan pada pelatihan model. Pada pengujian performa menggunakan fungsi $tune(method, data, kernel, ranges)$ pada aplikasi *R Studio.* Hasil dari pengujian performa seperti pada tabel 4.14

Tabel 4.14 Hasil Pengujian Performa

no	*cost*	*gamma*	*error*	*dispersion*
1	0.00	0.5	0.223306	0.03931595
2	0.01	0.5	0.20918	0.0378771
3	0.10	0.5	0.17527	0.03092137
4	1.00	0.5	0.174222	0.01979974
5	10.00	0.5	0.17759	0.02325389
6	100.00	0.5	0.188008	0.02407398
7	0.00	1	0.223465	0.03921491
8	0.01	1	0.210353	0.03720122
9	0.10	1	0.175274	0.02867015
10	1.00	1	0.172715	0.01893523
11	10.00	1	0.178499	0.02011036
12	100.00	1	0.197285	0.02275147
13	0.00	2	0.223466	0.03916316
14	0.01	2	0.210471	0.03669553
15	0.10	2	0.175898	0.02831701
16	1.00	2	0.172553	0.0185049
17	10.00	2	0.181515	0.01990162
18	100.00	2	0.199552	0.02667398

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada tabel 4.14 dapat disimpulkan bahwa performa terbaik memiliki parameter *cost* sebesar 1.00 dan parameter *gamma* sebesar 2.00 dengan nilai performa sebesar 0.1725534.



Gambar 4.33 *Statistics Error Value*

Pada gambar 4.33 didapatkan dari tabel 4.14 dengan percobaan uji coba 1 sampai 18 terhadap *statistics error value.* Dari statistik tersebut disimpulkan bahwa *statistics error value* terendah adalah pada uji coba ke-10 dengan *error value* sebesar 0.1725534.

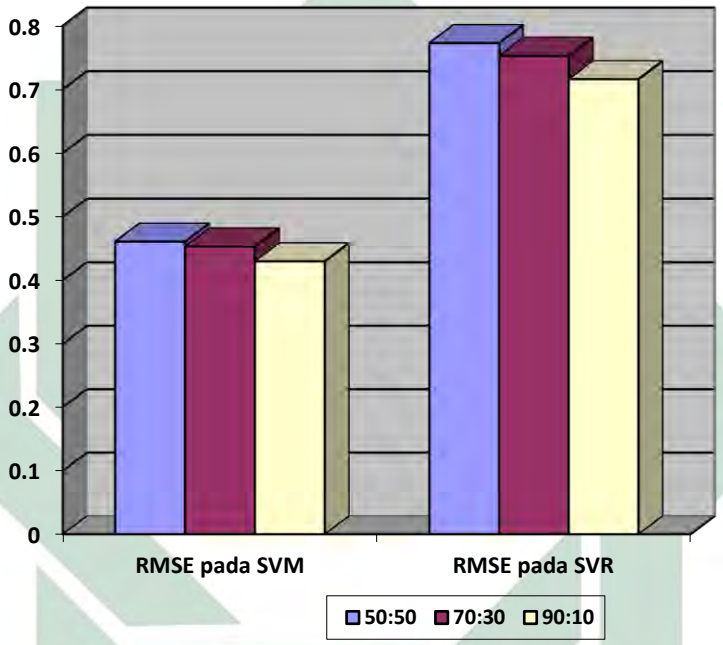
4.4.4 Perbandingan Metode

Pada penelitian ini menggunakan metode klasifikasi *Support Vector Machine* (SVM) dan *Support Vector Regression* (SVR), maka pada tahap ini dari kedua metode tersebut akan dibandingkan dengan hasil perbandingan pada tabel 4.15.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.15 Perbandingan Metode

Rasio Proporsi Data	RMSE pada SVM	RMSE pada SVR
50: 50	0.4608412	0.7726513
70: 30	0.4533824	0.7527727
90: 10	0.4299697	0.7159644



Gambar 4.34 *Statistics RMSE Value*

Pada gambar 4.34 didapatkan dari tabel 4.15 dengan uji coba rasio proporsi data terhadap *RMSE value.* Pada statistic tersebut dapat dilihat bahwa nilai perbandingan metode terbaik adalah pada rasio proporsi data 90: 10 dengan nilai RMSE pada SVM sebesar 0.4299697 dan nilai RMSE pada SVR sebesar 0.7159644. Dengan begitu dapat disimpulkan bahwa dalam kasus klasifikasi terbaik, metode SVM adalah metode yang lebih optimal dibandingkan dengan metode SVR.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan yang sudah dijelaskan tentang klasifikasi kinerja karyawan berbasis *Support Vector Machine* menggunakan *screen monitor durasion,* maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi *monitor screen durasion* dikembangkan dengan menerapkan teknologi *machine learning* yang menggunakan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi untuk dapat memprediksi kinerja karyawan. Dalam aplikasi ini diperlukan dua aplikasi, yaitu aplikasi dari sisi *client* dan *server*. Pada *server side*, aplikasi ini diintegrasikan dengan menerapkan *web sevice* yang menggunakan metode *Rest API*. Pada *client side* berupa aplikasi *desktop* yang dibangun dengan menggunakan framework *electron*. Aplikasi *desktop* tersebut melakukan pengiriman data dan pemanggilan *endpoint* dari *Rest API* yang diakses oleh *user* sehingga data-data akan masuk dalam *server side* yang kemudian akan diklasifikasi menjadi nilai kinerja karyawan. Pembangunan aplikasi dilakukan dengan serangkaian proses yang terdiri dari analisis, desain, implementasi dan pengujian.
2. Klasifikasi kinerja karyawan dihasilkan dengan menerapkan pembelajaran mesin dengan metode SVM. Metode SVM diterapkan dengan melakukan *preprocessing data* pada dataset yang menghasilkan enam variabel yaitu *no_of_workers, target_productivity, smv, wip, over_time* dan *actual_productivity.* Selanjutnya dilakukan proses pelatihan model dengan proses *training* dan proses *testing.* Pelatihan model ini menggunakan parameter terbaik yaitu *cost* sebesar 1.00, *gamma* sebesar 2.00 dan 90: 10 rasio proporsi data *testing* dan data *training.*
3. Pengujian aplikasi menggunakan metode *black box testing* dengan mendefinisikan *test case* terlebih dahulu kemudian melakukan pengujian terhadap *test case.* Hasil pengujian *black box testing* pada aplikasi *monitor screen duration* adalah memenuhi ekspektasi dari seluruh *test case* yang diujikan. Pada pengujian *Root Mean Squared Error* (RMSE) yaitu dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

melakukan pengujian pada masing-masing metode dengan menerapkan skenario pembagian rasio proporsi data *testing* dan data *training.* Hasil pengujian ini diperoleh perbandingan metode terbaik adalah pada rasio proporsi data $90:10$ dengan nilai RMSE pada SVM sebesar 0.4299697 dan nilai RMSE pada SVR sebesar 0.7159644.

5.2 Saran

Dari hasil penelitian ini masih jauh dari kesempurnaan, untuk menciptakan klasifikasi kinerja karyawan yang berkualitas tentu perlu dilakukan pengembangan pendekatan yang lainya, berikut beberapa saran untuk penelitian kedepannya:

1. Menambahkan pengujian kualitas aplikasi untuk mengetahui seberapa besar kualitas aplikasi yang telah dibuat.
2. Menambahkan metode pengujian *white box testing* secara menyeluruh pada setiap prosesnya untuk menguji aliran data, penangganan pengecualian dan tingkat kesalahan.
3. Memperbanyak uji coba parameter dan skenario pada pengujian performa supaya pada proses perhitungan pelatihan model dapat meperoleh hasil yang lebih akurat.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR PUSTAKA

Abubakar, A., Mantoro, T., Moedjiono, S., Ayu, M. A., Chiroma, H., Waqas, A., Abdulhamid, S. M., Hamza, M. F., & Gital, A. Y. u. (2016). A support vector machine classification of computational capabilities of 3D map on mobile device for navigation aid. *International Journal of Interactive Mobile Technologies*, *10*(3), 4–10. https://doi.org/10.3991/ijim.v10i3.5056

Akbulaev, N., Mammadov, I., & Aliyev, V. (2020). Economic Impact of COVID-19. In *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3649813

Alexander, F. R., Durai, P., & Keetticka, F. (2017). *Applicant Tracking System (Ats) a Component of Hris: Benefits and Issues in Its Usage*. *4*(11). https://en.wikipedia.org/wiki/Applicant_tracking_system

Bakhri, S., & Futiah, V. (2020). Pendampingan dan Pengembangan Manajemen Pemasaran Produk UMKM Melalui Teknologi Digital Di Masa Pandemi Covid-19. *Jurnal Loyalitas Sosial: Journal of Community Service in Humanities and Social Sciences*, *2*(2), 59. https://doi.org/10.32493/jls.v2i2.p59-70

Beunza, J. J., Puertas, E., García-Ovejero, E., Villalba, G., Condes, E., Koleva, G., Hurtado, C., & Landecho, M. F. (2019). Comparison of machine learning algorithms for clinical event prediction (risk of coronary heart disease). *Journal of Biomedical Informatics*, *97*(May), 103257. https://doi.org/10.1016/j.jbi.2019.103257

Bustamin, S. (2021). *Aplikasi Dekstop Multi Platform untuk Redis Client Framework Electron JS dan React JS*. *02*(01), 21–25.

Filipova, O. (n.d.). *Sof tware Development From A to Z*.

García-Peñalvo, F. (2020). Evaluación online : la tormenta perfecta. *Ensinar A Distância*, 2–4. https://bit.ly/2yO3K39

Goshu, Y. Y., Matebu, A., & Kitaw, D. (2017). Development of Productivity Measurement and Analysis Framework for Manufacturing Companies. *Journal of Optimization in Industrial Engineering*, *10*(22), 1–13. https://doi.org/10.22094/joie.2017.274

Huang, B., Cai, Z., Gu, Q., & Chen, C. (2008). Using support vector regression for classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5139 LNAI*, 581–588. https://doi.org/10.1007/978-3-540-88192-6_59

Ilag, B. N. (2021). Tools and Technology for Effective Remote Work. *International Journal of Computer Applications*, *174*(21), 13–16. https://doi.org/10.5120/ijca2021921109

ILO. (2021). Labour market developments. *Journal Labour Market Development*, *4*(7), 1–35.

Kemenkes, 2021. (2021). *IKHTISAR MINGGUAN COVID-19 Gambaran Situasi*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

nasional. *2021*(September), 1–21.

Khoza, L., & Marnewick, C. (2020). Waterfall and agile information system project success rates-a South African perspective. *South African Computer Journal*, *32*(1), 43–73. https://doi.org/10.18489/sacj.v32i1.683

Kowalczyk, A. (2017). Support Vector Machines Succintctly, Syncfusion. *Succinctly E-Book Series*, 114. www.syncfusion.com.

Lucas, P. J. F., & van der Gaag, L. C. (1991). Principles of Expert Systems. *Addison-Wesley Longman*, *January 1991*, 426. https://www.cs.ru.nl/~peterl/proe.pdf

Michael, T. (2017). Manajemen Sumber Daya Manusia, Edisi Revisi Kedua. *Yogyakarta: Penerbit BPFE-UGM*, 177.

Moore, A. D. (2000). Employee Monitoring and Computer Technology: Evaluative Surveillance V. Privacy. *Business Ethics Quarterly*, *10*(3), 697–709. https://doi.org/10.2307/3857899

Mustajab, D., Bauw, A., Rasyid, A., Irawan, A., Akbar, M. A., & Hamid, M. A. (2020). Working From Home Phenomenon As an Effort to Prevent COVID-19 Attacks and Its Impacts on Work Productivity. *TIJAB (The International Journal of Applied Business)*, *4*(1), 13. https://doi.org/10.20473/tijab.v4.i1.2020.13-21

Nanda, S. K., Ranjan Mohanty, R., Sukla, S., & Chandra Ghosh, G. (2011). Development of Intelligence Process Tracking System for Job Seekers. *International Journal of Managing Information Technology*, *3*(4), 57–70. https://doi.org/10.5121/ijmit.2011.3405

Osborne, S., & Hammoud, M. S. (2017). Effective Employee Engagement in the Workplace. *International Journal of Applied Management and Technology*, *16*(1), 50–67. https://doi.org/10.5590/ijamt.2017.16.1.04

Purwianti, L., & Dila, W. N. (2021). *Perancangan Dan Penerapan Strategi Marketing Mix Pada Umkm Tati ' s Cake*. *1*(1), 2368–2376.

Ren, T., Kang, X., Sun, W., & Song, H. (2018). Study of dynamometer cards identification based on root-mean-square error algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, *32*(2), 1–18. https://doi.org/10.1142/S0218001418500040

Setiati, S., & Azwar, M. K. (2020). COVID-19 and Indonesia. *Acta Medica Indonesiana*, *52*(1), 84–89.

Sommerville, I. (2011). *Software Engineering (9th ed.; Boston, Ed.). Massachusetts: Pearson Education.*

Time, E., On, M., Performance, E., & Company, N. (2020). *I j r s m*. *7*(1), 72–82. https://doi.org/10.5281/zenodo.3612089

Veiga, A. K., Saraiva, A. M., Chapman, A. D., Morris, P. J., Gendreau, C., Schigel, D., & Robertson, T. J. (2017). A conceptual framework for quality assessment

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

and management of biodiversity data. *PLoS ONE*, *12*(6), 1–20. https://doi.org/10.1371/journal.pone.0178731

Wesley, P. A. (2015). Unified Modeling Language User Guide , The (2nd Edition) (Addison-Wesley Object Technology Series) Unified Modeling Language User Guide , The Unified Modeling Language User Guide , The Many of the designations used by manufacturers and sellers to dist. In *ResearchGate* (Vol. 2nd, Issue August).

WHO - World Health Organization. (2020). Considerations for school-related public health measures in the context of COVID-19. *World Health Organisation*, *2020*, 1–10. https://www.who.int/publications-detail/risk-

Žnidaršic, J., & Bernik, M. (2021). Impact of work-family balance results on employee work engagement within the organization: The case of Slovenia. *PLoS ONE*, *16*(1 January), 1–17. https://doi.org/10.1371/journal.pone.0245078

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id