

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu dalam penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Jurnal / Conference	Judul	Penulis	Hasil	Yang Diadopsi
International Journal of Electrical and Computer Engineering, Vol. 14 No. 6, 2024 [9]	<i>Faculty Assistant Bot-automation of Administrative Activities using Robotic Process Automation</i>	Mamidyala Durga Prasad, Balusu Nandini	Sebuah sistem RPA yang mampu menyimpan data dan memperbarui laporan hasil ujian mahasiswa di Universitas Telagana, India.	Konsep pengambilan data dan pembuatan laporan (<i>report</i>) secara otomatis dengan menggunakan RPA, serta <i>tool</i> yang digunakan untuk membangun sistem RPA, yaitu UiPath.
2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), 2022 [10]	<i>Monitoring and Facilitating Students Programming Skill Development using Robotic Process Automation (RPA) and Artificial Intelligence (AI)–A Case Study</i>	M. Somasundaram, V. Sumitra, Reddy S P Vijaya Vardan, B. Sakthipriya, K. Pavithra, M.K. Nivedha	Sebuah sistem terautomasi untuk pembuatan laporan kinerja siswa dalam sebuah program <i>training programming online</i> .	Konsep automasi pembuatan <i>report</i> dengan menggunakan sistem RPA.
<i>Electronics</i> , Vol 12 No. 8, 2023 [11]	<i>Implications for Sustainability Accounting and Reporting in the Context of the Automation-Driven</i>	Valentin Florentin Dumitru, Bogdan-S tefan Ionescu, Sînziana-Maria Rîndas, Laura-Eugenia-Lavinia Barna,	Sebuah sistem ERP terautomasi untuk mengerjakan keperluan terkait akuntansi beserta pembuatan pelaporannya.	<i>Tool</i> yang digunakan untuk membangun sistem RPA, yaitu UiPath

Jurnal / Conference	Judul	Penulis	Hasil	Yang Diadopsi
	<i>Evolution of ERP Systems</i>	Alexandru-Mihai Crişman		
International Journal of Research and Analytical Reviews, Vol. 11 No. 1, 2024 [19]	<i>Comparing the Experience of RPA Implementation in Public & Private Sector Organizations- A Case of an RPA Consulting Company</i>	Ifrah Bukhari	Sebuah perbandingan dalam implementasi RPA pada perusahaan sektor publik maupun swasta. Perusahaan sektor swasta lebih terbuka terhadap perubahan, memiliki pemahaman yang lebih baik tentang teknologi, dan lebih kooperatif dalam mendukung implementasi teknologi baru seperti RPA.	Penerapan RPA pada perusahaan swasta guna mengurangi biaya (<i>cost reduction</i>)
Educational Administration: Theory and Practice, Vol. 30 No. 1, 2024 [20]	<i>Robotic Process Automation (RPA) In Business Operations: Opportunities And Implementation Strategies</i>	Dr. Nirvikar katiyar, Dr. Richa Mishra, Mr. Shubham Chaurasia, Sabiya fatima, Nigar Siddiqui, Shreyi Mittal Saxena	Hasil eksplorasi peran RPA dalam operasi bisnis, yang membahas peluang, tantangan, dan strategi implementasinya.	Penerapan RPA karena peluang yang diberikannya yaitu, meningkatkan efisiensi, pengurangan biaya, dan peningkatan akurasi dari pekerjaan yang dikerjakan.
Jurnal Informatika dan Teknologi Komputer, Vol. 3 No. 3, 2023 [13]	Sistem Informasi Manajemen Dokumen Menggunakan Metode Prototyping Pada PT. Bangun Indah Saputra	Daffa Bayu Prayogaa, Dian Nur Sholihaningtias, Endaryono	Sebuah <i>website</i> yang membantu proses manajemen dokumen.	Metode <i>prototyping</i> yang digunakan untuk membangun <i>website</i> .
Hexagon, Vol. 2 No. 2, 2021 [14]	Implementasi Metode <i>Prototyping</i> Pada Sistem	Dony Ardiyansah, Omar Pahlevi, Tri Santoso	Sebuah sistem untuk pengadaan barang cetakan	Metode <i>prototyping</i> yang digunakan

Jurnal / Conference	Judul	Penulis	Hasil	Yang Diadopsi
	Informasi Pengadaan Barang Cetak Berbasis Web		pada PT. Bank ICBC.	untuk membangun <i>website</i> .
NARATIF (Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika), Vol. 3 No. 1, 2021 [17]	Sistem Informasi Manajemen Gudang Obat Menggunakan <i>Database</i> dan <i>Form Oracle</i> Di PT. Leuwitex	Aditya Sudiatmo	Sebuah aplikasi manajemen gudang obat yang menggunakan Oracle.	Oracle sebagai <i>database</i> yang digunakan dalam penelitian ini.
Journal of Informatics and Communications Technology, Vol. 3 No. 2, 2021 [18]	Integrasi <i>Learning Management System</i> dan <i>Database</i> Eksternal Menggunakan Oracle Studi Kasus: IT Telkom Jakarta	Demi Adidrana, Deny Haryadi, Seandy Arandiant Rozano	Sebuah <i>Learning Management System</i> yang diintegrasikan dengan <i>Database</i> Oracle.	Oracle sebagai <i>database</i> yang digunakan dalam penelitian ini.
Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika, Vol. 15 No. 1, 2021 [21]	Perancangan Desain Basis Data Sistem Informasi Geografis Tanah Penduduk Dengan Menerapkan Model Data Relasional (Studi Kasus: Desa Tumbang Mantuhe Kabupaten Gunung Mas Provinsi Kalimantan Tengah)	Tri Amri Wijayaa, Constantin Mentenga, Adi Suryaa, Afis Juliantoa, Ema Utami	Rancangan <i>relational database</i> untuk sistem informasi geografis tanah penduduk.	Tahap perancangan <i>relational database</i> untuk menyimpan data yang akan tampil pada <i>preventive maintenance report</i> .

Dari kesepuluh penelitian terdahulu yang disebutkan dalam Tabel 2.1, bahwa selain dari kebutuhan PT XYZ, penelitian ini mengambil referensi dari penelitian terdahulu. Konsep dasar dari penelitian ini, yaitu untuk melakukan automasi dalam pembuatan sebuah *report* diambil dari 1 jurnal dan 1 *conference paper*. Jurnal yang digunakan berjudul “*Faculty Assistant Bot-automation of Administrative Activities using Robotic Process Automation*” oleh Mamidyala

Durga Prasad dan Balusu Nandini, yang dimana jurnal ini menghasilkan sebuah sistem RPA yang mampu menyimpan data dan memperbarui laporan hasil ujian mahasiswa di Universitas Telagana, India. *Conference paper* yang digunakan berjudul “*Monitoring and Facilitating Students Programming Skill Development using Robotic Process Automation (RPA) and Artificial Intelligence (AI)—A Case Study*” oleh M. Somasundaram, V. Sumitra, Reddy S P Vijaya Vardan, B. Sakthipriya, K. Pavithra, dan M.K. Nivedha yang dimana *paper* ini menghasilkan sistem terautomasi untuk pembuatan laporan kinerja siswa dalam sebuah program *training programming online*. Terdapat juga jurnal untuk penerapan RPA pada perusahaan sektor swasta guna mengurangi biaya perusahaan, yang berjudul “*Comparing the Experience of RPA Implementation in Public & Private Sector Organizations- A Case of an RPA Consulting Company*” oleh Ifrah Bukhari. Dan juga, terdapat jurnal referensi peluang dari penerapan RPA yang mampu meningkatkan efisiensi, pengurangan biaya, dan peningkatan akurasi dari pekerjaan yang dikerjakan, yaitu dari jurnal yang berjudul “*Robotic Process Automation (RPA) In Business Operations: Opportunities And Implementation Strategies*” oleh Dr. Nirvikar katiyar, Dr. Richa Mishra, Mr. Shubham Chaurasia, Sabiya fatima, Nigar Siddiqui, dan Shreyi Mittal Saxena.

Tidak hanya konsep dasar, terdapat juga 1 *tool* yang digunakan dalam penelitian ini, yaitu UiPath untuk membangun sistem RPA. *Tool* pertama, terdapat UiPath yang mengambil inspirasi dari 2 penelitian terdahulu. Pertama, terdapat jurnal dengan judul “*Faculty Assistant Bot-automation of Administrative Activities using Robotic Process Automation*” yang menghasilkan sistem RPA dengan menggunakan UiPath. Kedua, terdapat jurnal berjudul “*Implications for Sustainability Accounting and Reporting in the Context of the Automation-Driven Evolution of ERP Systems*” oleh Valentin Florentin Dumitru, Bogdan-Stefan Ionescu, Sînziana-Maria Rîndas, Laura-Eugenia-Lavinia Barna, dan Alexandru-Mihai Crîjman yang menghasilkan sebuah sistem RPA untuk menyelesaikan urusan akuntansi serta pembuatan pelaporannya dengan menggunakan UiPath.

Terdapat juga *database* yang digunakan dalam penelitian ini, yaitu Oracle. Penggunaan *Database* Oracle mengambil inspirasi dari 2 penelitian terdahulu. Penelitian pertama berupa jurnal berjudul “Sistem Informasi Manajemen Gudang Obat Menggunakan *Database* dan *Form* Oracle Di PT. Leuwitex” oleh Aditya Sudiatmo, yang dimana penelitian ini menghasilkan sebuah aplikasi manajemen gudang obat yang menggunakan Oracle. Penelitian kedua berbentuk jurnal yang berjudul “Integrasi Learning Management System dan Database Eksternal Menggunakan Oracle Studi Kasus: IT Telkom Jakarta” oleh Demi Adidrana, Deny Haryadi, dan Seandy Arandiant Rozano. Penelitian ini menghasilkan sebuah *Learning Management System* yang diintegrasikan dengan *Database* Oracle.

Selain konsep dasar dan *tool*, metode pengembangan sistem dalam penelitian ini juga terinspirasi dari 2 penelitian terdahulu. Penelitian pertama berupa jurnal yang berjudul “Sistem Informasi Manajemen Dokumen Menggunakan Metode Prototyping Pada PT. Bangun Indah Saputra” oleh Daffa Bayu Prayogaa, Dian Nur Sholihaningtias, dan Endaryono, yang dimana penelitian ini menghasilkan sebuah sistem *website* untuk manajemen dokumen. Penelitian kedua berupa jurnal yang berjudul “Implementasi Metode *Prototyping* Pada Sistem Informasi Pengadaan Barang Cetak Berbasis Web” oleh Dony Ardiyansah, Omar Pahlevi, dan Tri Santoso. Penelitian kedua ini menghasilkan sebuah *website* yang membantu PT. Bank ICBC dalam menyelesaikan permasalahan *procurement*, yang dimana masih kurangnya kontrol pada pencatatan barang.

Melalui analisis yang mendalam terhadap berbagai penelitian terdahulu, penelitian ini berhasil mengidentifikasi celah yang belum dieksplorasi, yaitu automasi pembuatan *preventive maintenance report*. Inovasi dari penelitian ini terletak pada pemanfaatan RPA untuk membuat sebuah *preventive maintenance report* tanpa campur tangan manusia, yang diharapkan dapat memberikan kontribusi signifikan terhadap pemahaman PT XYZ tentang RPA. Dengan demikian, hasil penelitian ini tidak hanya memperkaya khazanah ilmu pengetahuan, tetapi juga membuka peluang untuk penelitian lebih lanjut di

bidang ini, yang pada gilirannya dapat memberikan solusi yang lebih efektif terhadap permasalahan PT XYZ dalam pembuatan *preventive maintenance report* yang masih dilakukan secara manual.

2.2 Teori Topik Skripsi

2.2.1 Robot Process Automation (RPA)

Robot Process Automation atau yang disingkat sebagai RPA merupakan sebuah teknologi terautomasi berbasis *software* yang mampu untuk meniru tindak laku manusia untuk melakukan pekerjaan yang bersifat repetitif dan *non-value added*, seperti mengetik, menyalin, mengekstrak, dan memindahkan data dari suatu sistem ke sistem lain [22]. Pekerjaan yang dijalankan oleh RPA umumnya bersifat *rule-based*, yaitu pekerjaan tersebut dijalankan berdasarkan sekumpulan aturan yang telah ditetapkan, dan *well-structured*, yaitu pekerjaan tersebut dijalankan sesuai dengan urutan yang telah ditetapkan [23]. RPA telah digunakan pada beberapa bidang, seperti bidang asuransi, IT, *human resource*, dan sebagainya. Dalam menerapkan RPA, terdapat banyak tantangannya, salah satunya yang paling penting adalah dalam memilih proses yang tepat dalam mengimplementasikan RPA. Jika proses yang dipilih salah, maka sistem akan menjadi tidak efisien, bahkan dapat mengalami *failure* [24]. Terdapat beberapa keuntungan yang didapat perusahaan dalam menerapkan RPA, yaitu bisa menghemat biaya, waktu pengerjaan pekerjaan, serta meminimalisir terjadinya *human error* [25].

2.2.2 Basis Data (Database)

Basis data, atau yang umum disebut sebagai *database*, merupakan gabungan banyak data yang disimpan bersamaan dengan redundansi terkontrol guna melayani 1 aplikasi atau lebih dengan optimal. Dalam *database*, data dapat ditambahkan, dihapus, diubah, dan diambil secara terkontrol [26]. Terdapat 2 jenis *database*, yaitu *database* relasional dan non-relasional, atau yang sering dianggap sama dengan *database* NoSQL (*Not Only SQL*) [27]. Keduanya sudah banyak digunakan saat ini, dan kegunaannya bisa disesuaikan dengan keperluan.

2.2.3 Basis Data Relasional (*Relational Database*)

Database relasional merupakan jenis *database* yang menyimpan dan mengatur data yang tersimpan dengan struktur tabel, yaitu berupa kolom dan baris. *Database* jenis ini diciptakan berdasarkan model relasional yang digagaskan oleh Edgar F. Codd pada tahun 1970, dimana model ini menghubungkan antar tabel dengan menggunakan kunci (*key*). Setiap tabel dalam *database* relasional merupakan konsep atau entitas, sedangkan kolom merupakan properti entitas, dan baris merupakan *instance* atau rekaman dari masing-masing entitas. Terdapat 2 jenis *key* dalam *database* jenis ini, yaitu kunci primer (*primary key*) dan kunci asing (*foreign key*) yang berfungsi menjaga integritas data didalamnya. *Database* ini menggunakan SQL (*Structured Query Language*) yang berfungsi untuk melakukan mengambil, menambahkan, menghapus, dan mengubah data yang diinginkan [28].

2.2.4 Laporan Pemeliharaan Preventif Basis Data (*Database Preventive Maintenance Report*)

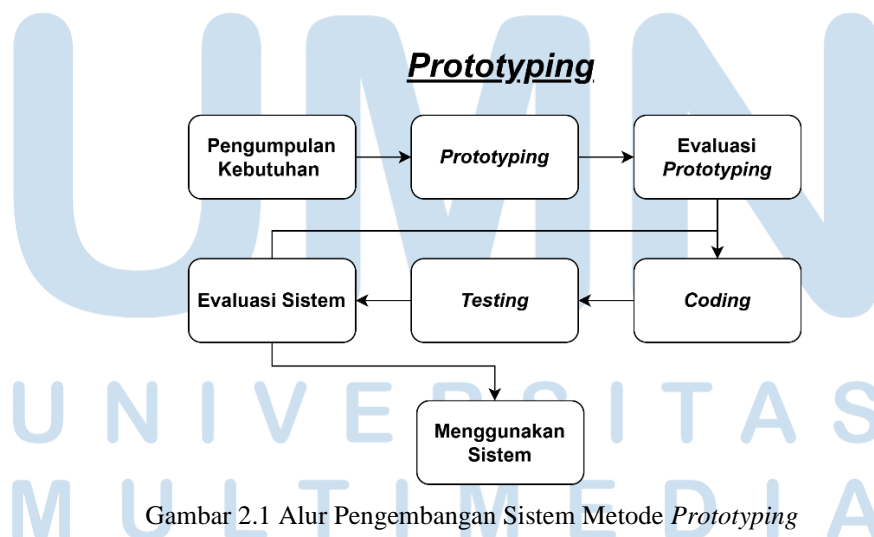
Menurut KBBI, berikut merupakan arti dari kata “laporan” adalah segala sesuatu yang dilaporkan, atau berita [29], sedangkan “pemeliharaan preventif” itu sendiri merupakan tindakan menjaga atau memelihara yang bertujuan untuk mencegah terjadinya masalah potensial [30]. Maka, ketika kedua definisi ini digabungkan, didapat sebuah definisi “laporan pemeliharaan preventif” yaitu sebuah berita yang isinya bersifat menjaga atau memelihara sesuatu dengan tujuan untuk mencegah terjadinya masalah potensial. Berdasarkan definisi ini serta definisi dari “basis data” atau “*database*”, dapat disimpulkan bahwa laporan pemeliharaan preventif basis data, yang dalam Bahasa Inggris disebut *database preventive maintenance report* merupakan sebuah berita yang isinya bersifat menjaga atau memelihara *database* dengan tujuan untuk mencegah terjadinya masalah potensial pada *database* tersebut. Pada perusahaan *IT consultant*, *database preventive maintenance report* ini dibuat setelah dilakukan *preventive maintenance*. Frekuensi dari *preventive maintenance* ini sendiri disesuaikan dengan kontrak perusahaan dengan klien, dan

umumnya dilakukan sekali setiap bulan. Isi dari report ini secara umum mengenai penggunaan *resource* dari *database*, *database health*, serta beberapa pengaturan penting, seperti *backup*, *disaster recovery*, *alert*, dan pengaturan lainnya yang diterapkan pada *database* tersebut selama 1 periode *preventive maintenance*.

2.3 Framework

Framework yang digunakan dalam penelitian ini adalah *framework prototyping* merupakan sebuah metode guna mendapatkan informasi yang dibutuhkan dengan cepat [31]. Model ini digunakan sebagai alat untuk memberikan gambaran kepada *user* (pengguna sistem) mengenai sebuah sistem berjalan. Pertama-tama, *user* akan diberi rancangan sistem yang akan dibuat. Setelah mendapatkan persetujuan, barulah sistem dikembangkan. Berikut beberapa kelebihan yang dimiliki dari metode ini:

1. *User* turut berpartisipasi aktif dalam proses mengembangkan sistem, dan hasil sistem akan sesuai dengan ekspektasi *user*.
2. Waktu pengembangan sistem menjadi lebih singkat
3. Komunikasi antara pengembang dan *user* dapat berjalan baik
4. Proses pengembangan sistem menjadi lebih mudah dan lebih sesuai dengan kebutuhan *user*.



Gambar 2.1 Alur Pengembangan Sistem Metode *Prototyping*

Terdapat pula penjelasan dari tahap-tahap pengembangan sistem dengan metode ini, yaitu [32]:

1. Pengumpulan Kebutuhan → Dilakukan identifikasi kebutuhan sistem.

2. *Prototyping* → Dilakukan perancangan sistem sementara.
3. Evaluasi *Prototyping* → Dilakukan evaluasi atas rancangan sistem sementara yang telah dibuat pada tahap sebelumnya.
4. Mengkodekan Sistem (*Coding*) → Merupakan tahap eksekusi atas *prototype* yang telah disetujui oleh *end user*.
5. Pengujian Sistem (*Testing*) → Dilakukan pengujian atas sistem yang telah selesai dibuat.
6. Evaluasi Sistem → Merupakan tahap evaluasi yang dilakukan oleh *end user* untuk menentukan apakah sistem boleh digunakan atau belum.
7. Menggunakan Sistem → Sistem yang telah berhasil melewati tahap sebelumnya dapat digunakan oleh *end user*.

2.4 Tools yang Digunakan

2.4.1 YugabyteDB

YugabyteDB merupakan *cloud-native distributed database* yang 100% *open source*. YugabyteDB ditemukan pada Februari 2016, oleh 3 orang *engineer* Facebook, yaitu Kannan Muthukkaruppan, Karthik Ranganathan, dan Mikhail Bautin [33]. YugabyteDB mendukung 2 jenis *client API* yaitu YSQL (*Yugabyte Structured Query Language*) dan YCQL (*Yugabyte Cloud Query Language*). Yugabyte YSQL kompatibel dengan PostgreSQL dan cocok digunakan untuk RDBMS (*Relational Database Management System*) *workload*, sedangkan YCQL kompatibel dengan Cassandra dan merupakan cocok untuk proses yang perlu memproses data dengan cepat seperti OLTP (*Online Transaction Processing*) dan HTAP (*Hybrid Transactional/Analytical Processing*) [34].

Berikut merupakan beberapa fitur yang di-*highlight* oleh Yugabyte [35]:

1. SQL dan *ACID transaction*

YugaByteDB mampu menjalankan operasi JOIN dan transaksi terdistribusi (*distributed transactions*) yang melibatkan banyak baris data di berbagai lokasi penyimpanan secara bersamaan, bahkan dalam skala besar. YugabyteDB juga mampu menyimpan data

dalam format dokumen dengan jaminan konsistensi, dan pemulihan otomatis.

2. *High performance dan massive scalability*

YugaByteDB memberikan *low latency* untuk aplikasi yang tersebar geografis (*geo-distributed application*) dengan berbagai tingkat *read consistency* dan replika data, serta kemampuan untuk meningkatkan kapasitas pemrosesan data secara *linear (linear-scalability)* seiring pertumbuhan data.

3. *Global data consistency*

YugaByteDB menyimpan data secara global dengan tetap menjaga konsistensi data. YugaByteDB memiliki fitur *auto-sharding* dan *auto-rebalancing* untuk membagi beban kerja secara rata ke seluruh *node*, bahkan dalam *cluster* yang besar.

4. *Built-in enterprise features*

YugaByteDB menawarkan fitur khusus untuk *enterprise*, seperti *distributed backups*, *data encryption*, *read replica*, *change data capture*, yaitu fitur untuk merekam perubahan data pada suatu *database* secara *real-time* dengan menggunakan *log file* dan *2 data center deployment*.

2.4.2 Yugabyte Anywhere

Yugabyte Anywhere atau sering disingkat sebagai YBA, merupakan sebuah *database-as-a-service (DBaaS)* yang membantu klien *manage YugabyteDB universe*. Sebuah *universe* dalam YBA hanya memiliki 1 *primary cluster*, sehingga hanya memiliki 1 *read-replica* yang terhubung dengan *primary cluster* tersebut. YBA dapat dijalankan pada 3 jenis *environment*, yaitu:

1. *On-premise* → YBA memungkinkan YugabyteDB dijalankan dalam VM (*Virtual Machine*) atau *bare metal* yang menggunakan berbagai jenis *Linux flavor*.

2. *Public cloud* → YBA memungkinkan YugabyteDB dijalankan pada berbagai jenis *cloud* seperti AWS, Azure, dan GCP (Google Cloud Platform).
3. Kubernetes → YBA memungkinkan *single available zone* YugabyteDB dijalankan dalam *single kubernetes cluster*. YBA memungkinkan *multi region* YugabyteDB dijalankan dalam *multi kubernetes cluster*.

YBA mendukung beberapa fitur seperti *encryption in transit*, *encryption at rest*, *backup* terjadwal, *alerting* dan *monitoring*, integrasi dengan LDAP dan OIDC untuk autentikasi, konfigurasi *high availability* [36].

2.4.3 UiPath

UiPath merupakan sebuah *platform* gratis yang digunakan untuk membuat RPA [37]. UiPath ditemukan pada tahun 2005 di Bucharest, Romania [38]. UiPath menggunakan arsitektur berbasis *website* yang dikembangkan menggunakan *.NET framework*. Terdapat 3 komponen utama dari UiPath, yaitu UiPath studio, UiPath Orchestrator dan UiPath Robots. UiPath mudah digunakan karena *user* hanya perlu melakukan *drag and drop* pada *platform*. Dalam pembuatan RPA, UiPath membantu dalam mengelola antrian tugas yang harus diproses dalam sistem RPA yang dibuat, memberikan sumber daya untuk menjalankan sistem, konfigurasi sistem, log aktivitas sistem, dan sebagainya. Terdapat 2 jenis RPA yang dapat dibuat oleh UiPath, yaitu *attended* dan *un-attended*. *Attended* berarti masih membutuhkan intervensi manusia untuk menyelesaikan pekerjaan dalam sistem RPA, sedangkan *un-attended* sebaliknya. UiPath menyediakan 5 jenis *recorder* dalam *platform*-nya untuk mempermudah proses automasi, yaitu *basic recording*, *desktop recording*, *web recording*, *image recording*, dan *citrix recording*. *Basic recording* digunakan untuk merekam aktivitas tunggal, seperti membuka atau menutup aplikasi, memilih kotak centang, dan sebagainya. *Desktop recording* digunakan untuk merekam beberapa aksi pada berbagai aplikasi *desktop*. *Web recording* digunakan untuk merekam aktivitas di aplikasi *web* dan *browser*.

Image recording dan *citrix recording* digunakan dalam virtual environment, seperti VM, dan dapat digunakan untuk automasi gambar, teks dan *keyboard* [39]. UiPath telah digunakan sebagai solusi dalam banyak industri, seperti keuangan dan perbankan, kesehatan, asuransi, manufaktur, dan sebagainya, serta telah digunakan dalam banyak departemen dalam perusahaan di industri tersebut, seperti keuangan, *quality assurance*, *contact center*, *human resource*, *supply chain*, dan departemen lainnya.

2.4.4 Oracle

Oracle merupakan sebuah *relational database* yang terdiri dari sekumpulan informasi dalam *Relational Database Management System* (RDBMS). Perusahaan Oracle menawarkan tipe basis data ini untuk berbagai aplikasi yang dapat beroperasi di berbagai platform dan merek perangkat keras. *Database* Oracle pertama kali diciptakan oleh Larry Ellison, Bob Miner, dan Ed Oates melalui perusahaan konsultasi yang mereka dirikan, Software Development Laboratories (SDL), pada tahun 1977. Kemudian, pada tahun 1983, perusahaan ini berganti nama menjadi Oracle Corporation dan masih eksis hingga kini [40].

Oracle merupakan produk *database* yang telah banyak digunakan perusahaan besar diseluruh dunia saat ini. Oracle memiliki keunggulan dalam sistem keamanannya, sehingga banyak dipilih untuk menunjang kebutuhan bisnis. Selain itu, Oracle juga mampu menyimpan dan mengelola data berkapasitas sangat besar, dan data-data tersebut dapat diakses dengan sangat cepat. Sintaks SQL yang sebagian besar mengikuti standar ANSI-92 memberikan kemudahan bagi pengembang *database* dalam merancang aplikasi, baik di sisi *front end* maupun *back end* [17].

2.4.5 PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP adalah bahasa pemrograman *script* yang dijalankan di server, biasanya digunakan untuk mengembangkan aplikasi *web* yang bersifat dinamis [41]. Bahasa

Pemrograman PHP umum digunakan untuk mengembangkan aplikasi berbasis *web*. Sebagai aplikasi, *website* tersebut harus memiliki sifat dinamis dan interaktif. Sifat dinamis berarti konten *website* dapat disesuaikan dengan kondisi tertentu. Sifat interaktif berarti *website* dapat memberikan umpan balik kepada *user*. PHP adalah bahasa pemrograman yang beroperasi di sisi server. Oleh karena itu, PHP akan diproses oleh server, dan hasil pemrosesannya akan dikirim kembali ke *browser* [42].



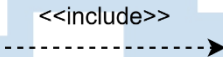
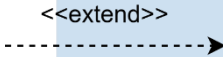


2.4.6 Unified Modeling Language (UML)

UML merupakan bahasa yang menggunakan diagram, gambar, atau grafik untuk menggambarkan, merancang, memvisualisasikan, serta mendokumentasikan proses pengembangan sistem [43]. UML umum digunakan di untuk mendefinisikan kebutuhan, melakukan analisis dan desain sistem, serta menggambarkan arsitektur sistem dalam konteks pemrograman [44]. UML menetapkan standar untuk penyusunan *blueprint* yang mencakup regulasi proses bisnis, skema *database*, *class* dalam bahasa pemrograman tertentu, dan berbagai komponen yang diperlukan dalam pembangunan sebuah sistem [43]. UML terdiri dari beberapa diagram visual, seperti *activity*, *use case*, dan *class diagram* [45].

1. Use Case Diagram

Use case diagram merupakan model yang menggambarkan bagaimana *user* (aktor) berinteraksi dengan sistem informasi. Dengan *use case*, kita dapat mengetahui fungsi-fungsi apa saja yang disediakan sistem dan siapa saja yang berwenang mengaksesnya [46]. Analisis hubungan antara *use case* dan aktor membantu dalam menentukan pihak-pihak yang memiliki otorisasi dalam menggunakan fungsionalitas yang terspesifikasi dalam *use case* [44]. Simbol serta notasi yang digunakan dalam *Use Case Diagram* dijelaskan pada Tabel 2.2.

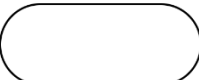


Tabel 2.2 Simbol dan Notasi pada *Use Case Diagram*


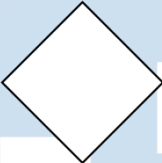

Simbol	Notasi
 Actor	<i>Actor</i>
	<i>Use Case</i>
	<i>Include</i>
	<i>Extension</i>
	<i>Association</i>
	<i>Generalization</i>

2. Activity Diagram

Activity diagram memberikan gambaran yang jelas mengenai urutan tindakan yang terjadi dalam suatu sistem atau proses bisnis [44]. *Activity Diagram* tidak hanya memodelkan aliran kerja, tetapi juga digunakan untuk mendefinisikan transisi antar *state* pada suatu sistem. Hubungan antar *state* ini direpresentasikan oleh panah yang menunjukkan arah aliran aktivitas [47]. Simbol serta notasi yang digunakan dalam *Activity Diagram* dijelaskan pada Tabel 2.3.

Tabel 2.3 Simbol dan Notasi pada *Activity Diagram*

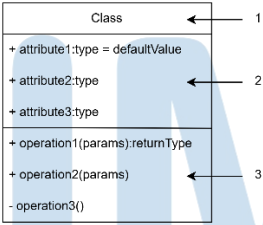




Simbol	Notasi
	<i>Activity</i>
	<i>Start of the Process</i>
	<i>Termination of the Process</i>

Simbol	Notasi
	<i>Synchronization Bar</i>
	<i>Decision Activity</i>
	<i>Initiate Activities</i>

3. Class Diagram

Class diagram berfungsi mengilustrasikan relasi antar *class* serta detail dari *class* kelas yang telah didefinisikan dalam desain sistem [44]. Diagram kelas berfungsi sebagai representasi visual yang mendetail mengenai struktur statis dari sistem, mencakup atribut, metode, dan hubungan antar *class* yang menjadi komponen utama dalam sistem tersebut [48]. Simbol serta notasi yang digunakan dalam *Activity Diagram* dijelaskan pada Tabel 2.3.

Tabel 2.4 Simbol dan Notasi pada *Class Diagram*

Simbol	Notasi
	<p><i>Class</i></p> <p>1 → <i>Class name</i></p> <p>2 → <i>Attributes</i></p> <p>3 → <i>Method</i></p>
	<i>Association</i>
	<i>Agregation</i>
	<i>Composition</i>
	<i>Generalization</i>

2.4.7 Entity Relationship Diagram (ERD)

ERD adalah diagram yang digunakan untuk membuat model konseptual dari sebuah *database*, sehingga kita dapat melihat hubungan

antar data secara jelas [49]. Model *entity relationship* memberikan gambaran lengkap tentang objek-objek dalam suatu sistem dan bagaimana objek-objek tersebut saling berhubungan, serta sifat-sifat yang melekat pada objek tersebut [50]. ERD digunakan untuk menggambarkan secara detail data yang ada dalam suatu sistem, termasuk batasan-batasannya. ERD terdiri dari tiga bagian utama yang saling berkaitan, yaitu [51]:

1. Entitas → Entitas merupakan sebuah unit data pokok yang ingin dipertahankan informasinya [52]. Entitas dapat berupa tempat, objek, peristiwa, atau orang yang penting untuk dicatat. Entitas memiliki atribut yang menggambarkan karakteristiknya. Terdapat dua jenis entitas utama dalam ERD, yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang dapat berdiri sendiri dan tidak bergantung pada entitas lain untuk keberadaannya. Sebaliknya, entitas lemah keberadaannya bergantung sepenuhnya pada entitas lain.
2. Atribut → Atribut merupakan kolom data dalam sebuah entitas [52]. Atribut berfungsi sebagai deskriptor yang mendefinisikan karakteristik spesifik dari setiap entitas dalam suatu *database*. Berdasarkan karakteristiknya, atribut dalam *database* dapat diklasifikasikan menjadi beberapa jenis, di antaranya adalah *simple attribute* dan *composite attribute*, *single value attribute* dan *multi value attribute*, *derived attribute*, *key attribute*.
3. Relasi → Relasi dalam *database* merujuk pada hubungan antara dua atau lebih entitas yang saling terkait. Relasi dapat dikategorikan menjadi beberapa jenis, yaitu *one-to one*, *many-to-many*, *many-to-one* dan *one-to-many*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A