

MEMBUAT APLIKASI PENGOLAHAN DATA ADMINISTRASI BARANG MENGGUNAKAN APLIKASI APEX ONLINE

MEMBUAT APLIKASI PENGOLAHAN DATA ADMINISTRASI BARANG MENGGUNAKAN APLIKASI APEX ONLINE

Syafril Fachri Pane, Wahyu Kurnia Sari, dan Zanwar Arif Wicaksono
D4 Teknik Informatika, Politeknik Pos Indonesia



Kreatif Industri Nusantara

Judul :

Membuat Aplikasi Pengolahan Data Administrasi Barang Menggunakan Aplikasi Apex Online

Penulis:

Syafril Fachrie Pane
Wahyu Kurnia Sari
Zanwar Arif Wicaksono

ISBN : 978-623-78980-3-0

Editor:

Wahyu Kurnia Sari
Zanwar Arif Wicaksono

Penyunting:

Syafril Fachrie Pane
Wahyu Kurnia Sari
Zanwar Arif Wicaksono

Desain sampul dan Tata letak:

Yogi Aditya Saputra

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2
Bandung 40191
Tel. 022 2045-8529
Email : awingga@kreatif.co.id

Distributor:

Informatics Research Center
Jl. Sariasisih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2020

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'*

Imam Syafi'i

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

SYAFRIAL FACHRI PANE, Dosen D4 Teknik Informatika Politeknik Pos Indonesia, Bandung, Indonesia

WAHYU KURNIA SARI, Mahasiswa D4 Teknik Informatika, Politeknik Pos Indonesia

ZANWAR ARIF WICAKSONO Mahasiswa D4 Teknik Informatika, Politeknik Pos Indonesia

CONTENTS IN BRIEF

1 PENGOLAHAN DATA ADMINISTRASI BARANG	1
2 HIRARKI JENJANG DATA	21
3 Latex	57
4 Model Basis Data	91
5 Rancangan Basis Data	99
6 Rancangan Basis Data	107
7 DATABASE SERVER	115
8 STUDI KASUS	127
9 Landasan Teori	131
10 Analisis dan Perancangan	137
11 Implementasi dan perancangan	145

DAFTAR ISI

Daftar Gambar	xv
Kata Pengantar	xxiii
Acknowledgments	xxv
Acronyms	xxvii
Glossary	xxix
List of Symbols	xxxi
Introduction	xxxiii
WAHYU KURNIA SARI DAN ZANWAR ARIF WICAKSONO	
1 PENGOLAHAN DATA ADMINISTRASI BARANG	1
1.1 Pengolahan Data Barang	1
1.2 <i>Oracle Apex Online</i>	2
1.3 Jenis Aplikasi yang dapat dibuat dengan <i>Application Express</i>	2
1.3.1 Architecture	2
1.3.2 Mengedit Aplikasi	3
1.3.3 Macam-macam Kunci Future	3
1.4 Konsep Dasar Aplikasi	4

1.5	Data	7
1.6	Sistem Informasi Berbasis Website	8
1.7	Algoritma	9
1.8	LaTeX	11
1.9	SQL	12
1.9.1	Data Definition Language (DDL)	12
1.9.2	Data Manipulation Language (DML)	13
1.9.3	Data Control Language (DCL)	13
1.9.4	Fungsi SQL	14
2	HIRARKI JENJANG DATA	21
2.1	Hirarki / Jenjang Data	21
2.2	Tahapan Instalasi Oracle Apex Offline	23
2.3	Pembuatan Tabel Pada SQL Command	41
2.3.1	To create a new table Using SQL Comand :	41
2.3.2	Show description Table :	42
2.3.3	Delete table Using SQL Comand :	43
2.3.4	Change table structure Using SQL Comand :	44
2.3.5	Delete table Using SQL Comand :	45
2.3.6	Insert, Update, and Delete Record dengan SQL Command	47
2.3.7	Delete Record	47
2.3.8	Membuat Table dengan Primer dan Komposit Key	48
2.4	Perancangan sistem	51
2.4.1	Macam - Macam Karakteristik Sistem	52
2.4.2	Klasifikasi Sistem	53
2.4.3	Analisis Sistem	53
2.4.4	Siklus Informasi	54
2.5	Sistem yang sedang berjalan	54
2.6	Dokumen yang digunakan	55
2.7	Sistem yang Dibangun	55
3	Latex	57
3.1	Latex	57
3.1.1	Tahapan Membuat Dokumen pada Latex	58
3.1.2	Pembukaan Dalam Pembuatan Dokumen	60
3.1.3	Menampilkan judul dokumen	61
3.1.4	Format Dasar: Abstrak, Paragraf, dan Baris baru	64
3.1.5	Kolom Komentar	65

3.1.6	Panduan referensi	67
3.1.7	Karakter	67
3.1.8	Memilih Kompiler LaTeX	68
3.1.9	Distribusi TeX	69
3.1.10	Editor Latex	69
3.1.11	Format Keluaran	70
3.1.12	Kompiler lain	71
3.1.13	Menyusun dokumen dengan referensi silang	72
3.1.14	Membuat dokumen di Overleaf	72
3.2	Panduan Cepat	79
3.2.1	Manajemen bibliografi di LaTeX	79
3.2.2	Penggunaan dasar	82
3.2.3	The bibliography file	85
3.2.4	Menyesuaikan daftar pustaka	87
3.2.5	Menambahkan daftar pustaka dalam daftar isi	89
4	Model Basis Data	91
4.0.1	Model Data Hirarki	93
4.0.2	Model data relasional	95
4.0.3	Fungsi Basis Data	97
4.0.4	Manfaat Basis Data	97
4.0.5	Komponen Basis Data	97
5	Rancangan Basis Data	99
5.0.1	Siklus Informasi	100
5.0.2	Sistem Basis Data dan Sistem File	100
5.0.3	Basis Data dan Sistem Basis Data (DBMS)	102
5.0.4	Komponen Sistem Basis Data	102
5.0.5	Abstraksi Data	103
5.0.6	Model Basis Data	103
5.0.7	Perancangan Database	104
6	Rancangan Basis Data	107
6.0.1	Siklus Informasi	108
6.0.2	Sistem Basis Data dan Sistem File	109
6.0.3	Basis Data dan Sistem Basis Data (DBMS)	110
6.0.4	Komponen Sistem Basis Data	111
6.0.5	Abstraksi Data	111

6.0.6	Model Basis Data	112
6.0.7	Perancangan Database	112
7	DATABASE SERVER	115
8	STUDI KASUS	127
8.1	Oracle Apex	127
8.2	Latar Belakang	127
8.3	Identifikasi Masalah	128
8.4	Tujuan Dan Manfaat	128
8.5	Ruang Lingkup	128
8.6	Sistematika Penulisan	128
9	Landasan Teori	131
9.1	Tinjauan Pustaka	131
9.2	Oracle Apex	132
9.3	Konsep Dasar Aplikasi	132
9.4	Sistem Informasi Berbasis Website	132
9.5	Web Mobile	132
9.6	Algoritma	132
9.7	Perancangan Sistem	133
9.8	Use Case	133
9.9	Activity Diagram	133
9.10	Sequence Diagram	133
10	Analisis dan Perancangan	137
10.1	Data	137
10.2	Sistem yang sedang berjalan	138
10.3	Dokumen yang digunakan	138
10.4	Sistem yang dibangun	138
10.5	Sistem yang Akan Dibangun pada penyampaian Informasi Terbaru	138
10.6	Kebutuhan Fungsional	139
10.7	Kebutuhan Perangkat Keras dan Perangkat Lunak	139
10.8	Kebutuhan Non-Fungsional	139
10.9	Use Case	140
10.10	Perancangan Antar Muka yang Dibangun	142
11	Implementasi dan perancangan	145

11.1	Lingkungan Implementasi	145
11.2	Lingkungan Perangkat keras	145
11.2.1	Langkah Kerja	146
11.2.2	LogBook	176
	Daftar Pustaka	201

DAFTAR GAMBAR

1.1	Fungsi SQL	14
1.2	Fungsi pada character	16
1.3	Fungsi pada character	17
1.4	Fungsi number	18
2.1	Hierarki/ Jenjang Data	21
2.2	Basis Data	22
2.3	Basis Data	23
2.4	File apex yang ada di Pc	23
2.5	Proses Penginstallan Aplikasi	24
2.6	Proses Penginstallan Aplikasi	25
2.7	Proses Penginstallan Aplikasi	26
2.8	Proses Penempatan File Aplikasi	27
2.9	Proses Penempatan File Aplikasi	28

2.10	Proses Pembuatan password Aplikasi	29
2.11	Review Installasi pada Aplikasi	30
2.12	Installasi Aplikasi	31
2.13	Installasi Aplikasi Telah Berhasil	32
2.14	CMD	33
2.15	CMD	34
2.16	Pengkonfigurasian pada CMD	35
2.17	Pengkonfigurasian pada CMD	36
2.18	Pengkonfigurasian	37
2.19	Pengkonfigurasian ke Database	38
2.20	Membuka User Anonymsos	39
2.21	Administrator pada aplikasi	40
2.22	Administrator pada aplikasi	41
2.23	Contoh Syntax Sql	41
2.24	Diskripsi pada tabel	42
2.25	Delete Pada Tabel	43
2.26	Alter Table	44
2.27	Alter Table	45
2.28	contoh	46
2.29	Insert Record	47
2.30	contoh	47
2.31	update	47
2.32	Contoh update	47
2.33	Delete	48
2.34	Contoh Delete	48
2.35	contoh perhatikanlah Skema Kampus	48
2.36	Contoh Tabel Mahasiswa	49
2.37	Contoh Tabel Matkul	49

2.38	Tabel Jadwal	49
2.39	Tabel Tipe Data	50
2.40	<i>Bentuk Umum Sistem</i>	51
2.41	<i>Siklus Informasi</i>	54
2.42	<i>Tabel Dokumen Yang Akan Digunakan</i>	55
2.43	<i>Diskripsi Perangkat Keras User</i>	56
2.44	<i>Diskripsi Perangkat Lunak User</i>	56
3.1	<i>Membuat dokumen pada lateks</i>	59
3.2	<i>Pembukaan dokumen pada lateks</i>	60
3.3	<i>Menampilkan dokumen pada lateks</i>	61
3.4	<i>Tampilan dokumen pada latex</i>	62
3.5	<i>Absbtrak paragraf baru</i>	64
3.6	<i>Contoh dari abstrak</i>	64
3.7	<i>Pembuatan pada kolom komentar 1</i>	65
3.8	<i>Hasil pemuatan kolom komentar</i>	66
3.9	<i>Panduan referensi dalam pembuatan lateks</i>	67
3.10	<i>Panduan referensi dalam karakter dalam lateks</i>	67
3.11	<i>Panduan referensi dalam karakter dalam lateks</i>	68
3.12	<i>Format keluaran dalam lateks</i>	71
3.13	<i>Penyusunan dokumen</i>	72
3.14	<i>User interface dalam lateks</i>	73
3.15	<i>User interface dalam login lateks</i>	73
3.16	<i>File yang tersimpan pada lateks</i>	74
3.17	<i>User interface dalam membuat proyek baru</i>	75
3.18	<i>Pembuatan objek baru</i>	76
3.19	<i>Editor dalam lateks</i>	77
3.20	<i>Contoh dokumen pada lateks</i>	78
3.21	<i>Contoh penggunaan Biblatex</i>	80

3.22	<i>Bibilatex</i>	80
3.23	<i>Parameter dasar dalam lateks</i>	82
3.24	<i>Penggunaan biblatex</i>	83
3.25	<i>syntax pada biblatex</i>	85
3.26	<i>Penyesuaian biblatex pada daftar pustaka</i>	87
3.27	<i>Penggunaan biblatex</i>	88
3.28	<i>parameter pada lateks</i>	89
3.29	<i>Penamahan daftar pustaka dalam daftar isi di biblatex</i>	89
3.30	<i>Tampilan setelah penggunaan biblatex di lateks</i>	90
4.1	<i>Contoh Data Flat</i>	92
4.2	Model data hirarki	93
4.3	Model data jaringan	94
4.4	Model data relasional	95
5.1	<i>Siklus Informasi</i>	100
6.1	<i>Siklus Informasi</i>	108
7.1	<i>Mysql</i>	116
7.2	<i>Oracle</i>	118
7.3	<i>Firebird</i>	120
7.4	<i>Sql server</i>	122
7.5	<i>Visual foxpro</i>	123
7.6	<i>paradox</i>	124
7.7	<i>postgre</i>	125
7.8	<i>Microsoft Access</i>	126
9.1	Use Case	134
9.2	Activity Diagram3	134
9.3	Activity Diagram3	135
10.1	Activity Diagram3	138
10.2	Perangkat User	139

10.3	Perangkat Lunak User	140
10.4	Use Case Sistem Administrasi Pengambilan Barang	140
10.5	Tampilan User Interface Login yang Telah dibuat	142
10.6	Tampilan User Interface Login yang Telah dibuat	143
11.1	Halaman Utama Pada apex oracle.com	146
11.2	<i>Form Request a Workspace</i>	147
11.3	Halaman Sign In Apex Oracle	147
11.4	Menu Pada Apex Oracle	148
11.5	Menu Pada SQL Workshop	149
11.6	Pembuatan Tabel Paket	150
11.7	Pembuatan Tabel Pegawai	151
11.8	Pembuatan Tabel Pegawai2	152
11.9	Data Pada Tabel Barang	153
11.10	<i>Trigger Auto Increment</i>	154
11.11	<i>Create Trigger Pegawai</i>	154
11.12	Create View	155
11.13	Menu <i>App Builder</i>	155
11.14	<i>Menu Create Application</i>	156
11.15	Tampilan membuat halaman aplikasi	157
11.16	<i>Add Page</i>	158
11.17	<i>Interactive Report</i>	159
11.18	<i>Select Table or View</i>	160
11.19	<i>Add Include Form</i>	161
11.20	Nama Setiap Halaman	161
11.21	<i>Add Page Dashboard</i>	162
11.22	<i>Add Page View</i>	163
11.23	<i>Create Application Button</i>	164
11.24	Tampilan Login Aplikasi Administrasi Barang	165

11.25	Tampilan Paket Masuk pada Aplikasi	165
11.26	<i>Edit Page</i> Paket Masuk	166
11.27	Paket Masuk Setelah diedit	166
11.28	Tampilan Paket Keluar pada Aplikasi	167
11.29	<i>Edit Page</i> Paket Keluar	167
11.30	Paket Keluar Setelah diedit	168
11.31	Form Pengisian Data Paket Masuk	169
11.32	<i>Edit Page</i> Form Paket Masuk	170
11.33	<i>Create Value of List</i>	171
11.34	<i>Input Value of List</i>	171
11.35	<i>Edit Value of List</i>	172
11.36	Form Pengisian Data Paket Masuk Setelah diedit	173
11.37	Tampilan Dashboard pada Laporan Grafik	173
11.38	<i>Edit Page Dashboard</i>	174
11.39	Tampilan <i>Dashboard</i> Hasil Edit	174
11.40	User Interface pada View	175
11.41	Pembuatan FK Sebelum Membuat Parent Key	176
11.42	Tabel Telah Dibuat	177
11.43	Tabel Pegawai	177
11.44	Tabel Pegawai Telah Dibuat	178
11.45	Tabel Kurir Error	178
11.46	Tabel Kurir Error	179
11.47	Pembuatan Tabel Pegawai	180
11.48	Combo Box	180
11.49	Tabel Barang	181
11.50	Create Trigger Tabel Pegawai2	181
11.51	Create Trigger Update Tabel Pegawai2	181
11.52	Create Trigger Dengan Fungsi Delete ke Pegawai2	182

11.53	Create Trigger dengan Insert,Update,Delete ke Pegawai	182
11.54	Percobaan Dengan Insert Menggunakan Aplikasi	182
11.55	Percobaan Tabel Pegawai	183
11.56	Percobaan Tabel Pegawai2	183
11.57	Update Pada Field Alamat	184
11.58	Sebelum Update	184
11.59	Setelah Update	184
11.60	Percobaan Delete	185
11.61	Data Pegawai2 Menjadi Kosong	185
11.62	Tampilan Paket Masuk Combo Box	186
11.63	Create list Of Values	186
11.64	Mengubah Menjadi Select List	187
11.65	Mengisi Value Sesuai List	187
11.66	Hasil Mengisi Value	188
11.67	Tampilan Paket Masuk	188
11.68	Tampilan Saat Merubah Query	189
11.69	Hasil Sistem Paket Masuk	189
11.70	Tampilan Menu Paket Keluar	190
11.71	Merubah Tampilan Table/View ke SQL Query	190
11.72	Tampilan Table View Menjadi SQL Query	191
11.73	Menambahkan Fungsi Status	191
11.74	Hasil Paket Keluar	192
11.75	Tampilan Menu Paket Masuk	192
11.76	Tampilan Perubahan Format Mask Pada Menu Apprance	193
11.77	Menu Paket	193
11.78	Penggantian pada appearances dengan memilih atribut	194
11.79	Hasil Paket Keluar	194
11.80	Tampilan Perubahan Format Mask Pada Menu Apprance	195

11.81	Tampilan Perubahan Format Mask Pada Menu Apprance	195
11.82	Tampilan Perubahan Format Mask Pada Menu Apprance	196
11.83	format tanggal	196
11.84	Solusi Format Mask	197
11.85	Penanganan error	198
11.86	Kurangnya Atribut Pengambilan Barang	199
11.87	Solusi Proses Pembuatan View	199
11.88	Grafik	200
11.89	Solusi Perubahan Atribut Chart Menjadi Bar	200

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan pembuatan aplikasi Proyek II ini, tanpa pertolongan-Nya mungkin penulis tidak akan sanggup menyelesaiannya dengan baik. Shalawat dan salam semoga terlimpah curahkan kepada Nabi Muhammad SAW beserta sahabat dan keluarga Beliau. Laporan ini disusun untuk memenuhi kelulusan matakuliah Proyek II pada Program Studi DIV Teknik Informatika. Proses Proyek II ini juga tidak terlepas dari bantuan pihak Pembimbing. Oleh karena itu, pada kata pengantar ini penulis menyampaikan terimakasih kepada :

1. Syafrial Fachri Pane, S.T., M.T.I. selaku Pembimbing Internal dan Penguji Utama dalam penyusunan buku ini :
2. Rolly Maulana Awangga, S.T., M.T. selaku Koordinator Proyek II Tahun Akademik 2019/2020;
3. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2019/2020
4. Dr. Ir. Agus Purnomo, M.T. selaku Direktur Politeknik Pos Indonesia Tahun Akademik 2019/2020[]

Penulis telah membuat laporan ini dengan sebaik-baiknya, diharapkan memberikan kritik dan saran dari semua pihak yang bersifat membangun, terimakasih.

Bandung, Jawa Barat

Februari, 2020

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari rekan - rekan mahasiswa serta dosen pembimbing agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk bapak SyafrialFachri Pane, S.T., M.T.I.,EBDP selaku pembimbing kami pada proyek 2 ini.

Wahyu Kurnia Sari dan Zanwar Arif Wicaksono

ACRONYMS

APEX Application Express

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

A Amplitude

$\&$ Propositional logic symbol

a Filter Coefficient

B Number of Beats

INTRODUCTION

WAHYU KURNIA SARI DAN ZANWAR ARIF WICAKSONO

D4 Teknik Informatika, Politeknik Pos Indonesia
Bandung, Jawa Barat, Indonesia

Apex oracle merupakan sebuah aplikasi database yang sering digunakan pada perusahaan besar khususnya menangani database.

BAB 1

PENGOLAHAN DATA ADMINISTRASI BARANG

1.1 Pengolahan Data Barang

Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. pengolahan data adalah suatu proses menerima dan mengeluarkan data menjadi bentuk lain yaitu berupa informasi. Pengolahan Data adalah manipulasi data agar menjadi bentuk yang lebih berguna. Pengolahan data tidak hanya berupa perhitungan numerik tetapi juga operasi-operasi seperti klasifikasi data dan perpindahan data dari satu tempat ke tempat lain. Menurut Siagianadministrasi adalah keseluruhan proses kerjasama antara dua orang manusia atau lebih yang didasarkan atas rasionalitas tertentu untuk mencapai tujuan yang telah ditentukan sebelumnya. Administrasi sebagai usaha dan kegiatan yang berkaitan dengan penyelenggaraan kebijakan untuk mencapai tujuan. Administrasi didefinisikan sebagai keseluruhan proses kerjasama antara dua orang manusia atau lebih yang didasarkan atas rasionalitas tertentu untuk mencapai tujuan yang telah ditentukan sebelumnya.

1.2 Oracle Apex Online

Oracle Application Express (Oracle APEX) yang dulu disebut *HTML-DB* adalah sebuah *framework* yang berbasis pada sebuah database *dedicated* (sementara ini sampai versi terbaru masih untuk *Oracle DB* saja dan lisensi *include* dalam lisensi *database*, ini artinya apa bahwa *engine* aplikasi dibangun sepenuhnya didalam sebuah *database* [1]). Oracle Apex ini mudah dipakai hanya menggunakan web browser saja dan proses pemrograman yang sederhana, serta mengembangkan ilmu dan kemampuan dengan Aplikasi ini secara aman dan cepat. Oracle Application Express memiliki kualitas database yang bagus, produktivitas dan standart luas yang dimiliki oleh perusahaan-perusahaan besar dan memiliki keamanan, stabilitas, ketersediaan dalam membangun suatu web. Application Express adalah suatu alat yang digunakan untuk membangun Aplikasi web-based. Selain itu Oracle Application Express tidak membutuhkan perangkat lainnya untuk mengembangkan, menyebarkan serta menjalankan Aplikasi, karena di dalam Oracle Application Express telah menyediakan tiga alat utama. Tiga alat utama ini memiliki Kegunaan yang penting di dalam Oracle Application Express:

1. Application Builder : Membuat aplikasi, melihat aplikasi, mengimport aplikasi, mengatur service, mengatur user aplikasi dan memantau aktifitas yang di lakukan pengguna.
2. SQL Workshop : Membuat tabel dan komponennya (menggunakan kode PL-SQL secara manual maupun otomatis), melihat struktur tabel dan komponennya, mengimpor dan mengekspor script.
3. Utilitas : melihat report table dan komponennya dan history aplikasi.

1.3 Jenis Aplikasi yang dapat dibuat dengan *Application Express*

Aplikasi *Express* digunakan untuk Aplikasi yang membuat report data pada database. Report data dapat berupa report text yang dikaitkan dengan report lainnya dan grafik supaya lebih efektif. *Application Express* juga bisa digunakan dalam pengeditan *database* meskipun memiliki kapasitas yang besar. Banyak bentuk tampilan yang bisa dibuat menggunakan *Application Express* antara lain: *radio group, checkboxes, select list, shuttles, text editor and date pickers*.

1.3.1 Architecture

Application Express merupakan aplikasi yang dimanfaatkan untuk mengambil data di dalam tabel database serta dapat juga membuat atau memodifikasi metadata yang disimpan di dalam database. Berikut ini grafik yang menggambarkan hubungan antara database Oracle, web browser, server Apache dan mod plsql.

1.3.2 Mengedit Aplikasi

Application Express dapat digunakan untuk mengubah Application Oracle form dan Microsoft access, selain itu juga menghasilkan Application Express dan dalam bentuk objek. Modernisasi format proyek Oracle sering dikerjakan untuk mengkonversi Aplikasi terdahulu untuk diwariskan (extend) kepada Application Oracle yang terakhir dan memungkinkan untuk berkembang.

1.3.3 Macam-macam Kunci Future

1.3.3.1 Report Application Oracle Express dapat menghasilkan report HTML, report ini dapat menampilkan hasil sql query serta dapat juga mendownload dan mencetak report didalam format HTML, PDF, RTF(tepat untuk Microsoft Word) dan XLS(tepat untuk Microsoft Excel).

1.3.3.2 Forms Oracle Apex menyediakan fasilitas untuk membuat form di table (secara otomatis) seperti untuk memasukan, memperbarui, menghapus data dan memperbarui serta mendeteksi data yang hilang. Materi form menawarkan berbagai pilihan yang mencakup radio group, checkbox, select list, shuttles, text editor dan date pickers.

1.3.3.3 Charting Charting di dalam Oracle Apex digunakan untuk pemanis tampilan pada Aplikasi serta mempermudah para pengguna untuk melihat hasil suatu data. Charting dapat diperbarui dengan PPR teknologi (Partial Page Refresh), untuk menghindari pembaruan seluruh halaman selain itu mengatur pembaruan table interval yang digunakan.

1.3.3.4 PDF Printing Application Express mampu untuk mengekspor suatu laporan ke PDF dan mencetak report. Serta menggambarkan query laporan dan tata ruang report dibawah komponen. Report itu dapat menjadi link dimanapun didalam sebuah Aplikasi.

1.3.3.5 Web Service Suatu sistem perangkat lunak yang dirancang untuk interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

1.3.3.6 Spreadsheet Upload Digunakan untuk membuat Aplikasi dari spreadsheet yang dapat mempercepat data upload spreadsheet secara langsung didalam database. Semua pengguna dapat memilih sumber data di tabel database baru atau menambahkan untuk mengeluarkan tabel database. Sekali data diupload pengguna dapat dengan cepat menulis Aplikasi.

1.3.3.7 Session State Management Application Express dengan mudah menyimpan konteks aplikasi didalam database. Pedoman didalam SQL dan PL/SQL sederhana menggunakan variabel, contohnya sebagai berikut:

```
SELECT * FROM EMP WHERE EMPNO = :P1 ID
```

Didalam contoh ini berisi item P1 ID secara otomatis terikat ketika query dijalankan. Serta dapat mengacu dengan awalan nama item dan akhiran itu dengan periode, contohnya sebagai berikut:

Untuk mengatur dari 2 dimensi pengaturan data, Oracle Application Express menyediakan koleksi infrakstruktur yang sempurna. Baik digunakan untuk melakukan manajemen semua sesi karena tidak memakan banyak memori

1.3.3.8 User Interface Themes Oracle Apex memisahkan pengguna penghubung dari Aplikasi, pengguna dapat mendesain aplikasi dengan satu tema. Memilih dari beberapa tema yang ada atau bisa membuat tema sendiri. Dengan begitu secara tidak langsung dapat memberi keuntungan kepada pengguna karena dapat menghemat waktu mereka.

1.4 Konsep Dasar Aplikasi

aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user [2]. Aplikasi pula merupakan sebuah program yang dibuat dengan tujuan untuk melaksanakan fungsi sesuai dengan kegunaan aplikasinya, penggunanya dan jenis aplikasi itu sendiri. Aplikasi dibuat dengan bahasa pemrograman yang bertujuan untuk membantu memecahkan masalah dengan aturan yang sesuai dengan bahasa pemrograman itu sendiri yang nantinya bisa mengolah data Pengertian aplikasi menurut para ahli adalah sebagai berikut :

1. Menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.
2. Menurut Kamus Besar Bahasa Indonesia (1998 : 52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.
3. Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows , permainan (game), dan sebagainya.
4. Menurut Harip Santoso, Aplikasi adalah suatu kelompok file (form, class, report) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset, dan lain-lain.
5. Menurut Ali Zaki dan Smitdev Community, pengertian aplikasi adalah suatu komponen yang bermanfaat sebagai media untuk mengoperasikan pengolahan

data atau kegiatan lainnya, misalnya; membuat dan mengolah file atau dokumen.

6. Menurut Sri Widiani, arti aplikasi adalah suatu perangkat lunak yang dibuat sebagai front end sebuah sistem yang dipakai untuk mengelola data sehingga menjadi suatu informasi yang bermanfaat bagi pengguna.

Maka dari itu suatu aplikasi memberikan kemudahan dan kenyamanan bagi manusia di berbagai bidang kehidupan. Adapun beberapa fungsi aplikasi adalah sebagai berikut:

1. Bidang Pendidikan Dalam bidang pendidikan aplikasi memiliki fungsi sebagai bahan pengajaran. Misalnya sebuah aplikasi yang berguna untuk penyajian materi yang dilengkapi animasi-animasi agar lebih menarik seperti Microsoft PowerPoint. Sehingga suasana pengajaran menjadi lebih nyaman dan dapat mencapai hasil yang lebih maksimal.
2. Bidang Kedokteran Peran aplikasi juga mencapai ke bidang kedokteran. Misalnya saja dengan adanya aplikasi maka dapat membantu dokter untuk mendiagnosa penyakit, meracik obat serta menawarkan perawatan rutin. Saat ini telah banyak dikembangkan aplikasi yang berkaitan dalam dunia medis.
3. Bidang Bisnis Dalam bidang bisnis, aplikasi memiliki fungsi untuk membantu menghitung besarnya keuntungan yang diperoleh. Karena dalam beberapa hal apabila dilakukan secara manual tentunya akan membutuhkan waktu yang lama sehingga dibutuhkan suatu aplikasi.
4. Bidang Ilmu Pengetahuan Dalam bidang ilmu pengetahuan aplikasi membantu untuk memperoleh informasi yang berkaitan dengan pengetahuan. Banyak aplikasi yang dikembangkan juga membantu dalam penelitian tertentu terkait ilmu pengetahuan. Selain itu, baru baru ini juga ada aplikasi yang dapat mempertemukan antara siswa yang ingin berkonsultasi kepada pengajar atau guru atau ahlinya.
5. Bidang Militer Ternyata aplikasi juga memiliki fungsi di bidang militer. Fungsi aplikasi di bidang militer bertujuan untuk melakukan pengontrolan pesawat sehingga tidak lagi dilakukan secara manual. Dengan demikian hasil yang diperoleh pun akan menjadi lebih optimal daripada menggunakan cara manual.

Untuk memudahkan dalam mengenal jenis-jenis aplikasi, maka aplikasi dapat dikelompokkan menjadi tujuh macam. Adapun pengelompokan aplikasi adalah sebagai berikut:

1. System Software, yaitu aplikasi yang dapat menglola dan mengendalikan proses operasi internal di dalam sistem komputer.
2. Real Time Software, yaitu aplikasi yang memiliki fungsi untuk mengamati, menganalisa, serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.

3. Business Software, yaitu aplikasi yang diciptakan dan dikembangkan untuk keperluan bisnis, misalnya untuk mengatur sistem keuangan.
4. Engineering and Scientific Software, yaitu aplikasi yang diciptakan dan dikembangkan untuk membantu manusia menyelesaikan suatu masalah yang sifatnya non algoritmik.
5. Web Based Software, yaitu aplikasi yang memiliki fungsi sebagai media penghubung antara pengguna dengan internet secara langsung.
6. Personal Computer Software, yaitu aplikasi yang digunakan untuk perangkat pengguna resmi dan juga pribadi yang sudah banyak digunakan saat ini.

1.5 Data

Data merupakan sekumpulan keterangan atau fakta mentah berupa simbol, angka, kata-kata, atau citra, yang didapatkan melalui proses pengamatan atau pencarian ke sumber-sumber tertentu. Definisi data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (observasi) dan dapat diolah menjadi bentuk yang lebih kompleks, seperti; informasi, database, atau solusi untuk masalah tertentu. Secara etimologis, istilah data berasal dari bahasa Latin, yaitu Datum yang artinya sesuatu yang diberikan. Dengan kata lain, data merupakan hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa simbol, warna, kata-kata, angka, atau citra. Adapun tentang data menurut para ahli sebagai berikut:

1. Menurut Arikunto Suharsimi, pengertian data adalah semua fakta dan angka-angka yang dapat dijadikan bahan untuk menyusun sebuah informasi.
2. Menurut Nuzulla Agustina, definisi data adalah keterangan mengenai suatu hal yang sudah sering terjadi dan berupa himpunan fakta, angka, grafik tabel, gambar, lambang, kata, huruf, yang menyatakan sesuatu pemikiran, objek, serta kondisi dan situasi.
3. Menurut Kuswadi dan E. Mutiara, pengertian data adalah sekumpulan keterangan yang diperoleh dari suatu pengamatan yang dapat berupa angka, lambang, atau sifat.
4. Menurut Slamet Riyadi, pengertian data adalah kumpulan keterangan yang diperoleh dari pengamatan dimana data bisa berupa angka-angka atau lambang-lambang.
5. Menurut Kristanto, pengertian data adalah suatu fakta mentah mengenai objek yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan dan kejadian.

Pada dasarnya data berhubungan dengan berbagai aspek kehidupan manusia. Namun, penggunaan istilah data lebih banyak dipakai di bidang teknologi komputer dan juga kegiatan penelitian.

Mengacu pada definisi data di atas, adapun fungsi data adalah sebagai berikut:

1. Data dapat berfungsi sebagai acuan dalam mengambil suatu keputusan dalam pemecahan masalah.
2. Data bisa dijadikan sebagai pedoman atau dasar suatu penelitian atau perencanaan.
3. Data dapat berfungsi sebagai acuan dalam implementasi suatu kegiatan.
4. Data berfungsi sebagai dasar untuk melakukan evaluasi terhadap suatu kegiatan.

1.6 Sistem Informasi Berbasis Website

suatu halaman web yang saling berhubungan yang umumnya berisikan kumpulan informasi berupa data teks, gambar, animasi, audio, video maupun gabungan dari semuanya yang biasanya dibuat untuk personal, organisasi dan perusahaan [3]. Dari pengertian website tersebut dapat dibedakan menjadi 2 yaitu web bersifat statis dan dinamis. Bersifat statis apabila isi informasinya tetap dan isi informasinya hanya dari pemilik website sedangkan web yang bersifat dinamis apabila isi informasinya selalu berubah-ubah dan dapat diubah-ubah oleh pemilik maupun pengguna website. Contoh web statis : website profil perusahaan, sedangkan contoh web dinamis seperti facebook, twitter dll.Terdapat 3 unsur yang sangat penting pada website. Tanpa adanya semua unsur ini, website anda tidak akan pernah ditemukan dan diakses oleh pengguna di internet. Ketiga unsur yang dimaksud adalah:

1. Domain. Jika website diibaratkan sebagai produk, maka domain adalah merk. Penggunaan domain yang menarik akan membuat orang tertarik untuk memasuki suatu website. Dengan pemilihan nama domain yang unik juga membuat orang mudah mengingatnya untuk nantinya dikunjungi kembali.
2. Hosting. Tidak kalah pentingnya dengan domain, hosting memiliki peran untuk menyimpan semua database (script, gambar, video, teks dan lain sebagainya) yang diperlukan untuk membentuk suatu website. Banyak sekali penyedia jasa hosting di Indonesia, salah satunya Niagahoster yang menyediakan hosting terbaik untuk kecepatan akses website Anda.
3. Konten. Tanpa adanya konten pada website, maka website bisa dikatakan tidak memiliki tujuan yang jelas. Konten pada website dapat berupa teks, gambar atau video. Jika dilihat dari konten yang disuguhkan, terdapat beberapa macam website. Misalnya saja, sosial media, website berita, website jual beli atau website yang berisi konten yang berdasarkan minat, bakat serta hobi.

Website merupakan media untuk menampilkan informasi. Awal kemunculannya, informasi yang ditampilkan pada website hanya berupa tulisan.Saat ini berbagai jenis website dapat ditemukan dengan mudah, tidak hanya menampilkan informasi berupa tulisan saja. Berikut adalah jenis-jenis website :

1. Website pribadi :

Saat ini tersedia berbagai layanan yang dapat Anda gunakan untuk membuat website pribadi. Mulai dari yang gratis sampai layanan berbayar.Untuk yang gratis, Anda harus bersedia memiliki alamat website seperti www.namalayanan.com/[username](http://www.namawebiste.blogspot.com) atau www.namawebiste.blogspot.com.Alamat website seperti contoh di atas tidak disarankan untuk Anda yang ingin memiliki konten lengkap, seperti dengan adanya video dan banyak gambar.Biasanya, penyedia layanan website gratis akan memberikan resource yang terbatas.Selain itu, untuk alamat dari website Anda akan sulit bersaing di mesin pencari. Oleh karena itu kami menyarankan Anda untuk membuat website melalui layanan penyedia domain dan

hosting. Sehingga Anda dapat membuat website yang lebih profesional dengan alamat www.namaanda.com.

2. Website Toko Online :

Toko online dengan fungsi yang tidak jauh berbeda dengan toko online yang ada saat ini? Jika benar, dengan menggunakan sebuah tool yang tepat Anda bisa membuatnya dalam hitungan jam bahkan menit, dan website toko online Anda sudah dapat diakses melalui internet.

3. Blog :

Pada umumnya blog berisi artikel yang bertujuan untuk berbagi pengetahuan, ide atau pengalaman penulis. Untuk contoh blog adalah apa yang saat ini Anda sedang baca, yaitu blog Niagahoster. Tujuan dari blog ini adalah untuk berbagi pengetahuan yang berhubungan dengan web hosting, internet marketing dan hal lainnya. Untuk artikel yang diterbitkan setiap blog berbeda-beda, mulai dari satu hari satu artikel atau bahkan satu artikel dalam satu minggu. Hal ini tergantung dari kualitas artikel yang dibuat dan jumlah penulisnya.

1.7 Algoritma

Pengertian algoritma adalah suatu urutan dari beberapa langkah logis dan sistematis yang digunakan untuk menyelesaikan masalah tertentu [1]. Pendapat lain mengatakan definisi algoritma adalah proses atau rangkaian aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah lainnya, terutama oleh komputer. Dengan kata lain, semua susunan logis yang diurutkan berdasarkan sistematika tertentu dan digunakan untuk memecahkan suatu masalah dapat disebut dengan algoritma. Algoritma digunakan untuk melakukan penghitungan, penalaran otomatis, serta mengolah data pada komputer dengan menggunakan software. Dalam algoritma terdapat rangkaian terbatas dari beberapa intruksi untuk menghitung suatu fungsi yang jika dieksekusi dan diproses akan menghasilkan output, lalu berhenti pada kondisi akhir yang sudah ditentukan. Tidak hanya dalam aplikasi, algoritma juga kita terapkan dalam kehidupan sehari-hari, seperti langkah-langkah dalam memasak, langkah-langkah dalam menyuci pakaian, dan masih banyak lagi. Algoritma yang akan diterapkan dalam pembuatan aplikasi ini adalah bagaimana cara terbentuknya aplikasi pengolahan data administrasi barang menggunakan aplikasi apex online. Adapun Pengertian Algoritma Menurut Para Ahli Agar lebih memahami apa itu algoritma, maka kita dapat merujuk pada pendapat para ahli berikut ini:

1. Menurut Abu Jafar Muhammad Ibnu Musa Al-Khawarizmi (ahli matematika dari Uzbekistan), pengertian algoritma adalah suatu metode khusus yang digunakan untuk menyelesaikan permasalahan.
2. Donald Ervin Knuth Menurut Donald Ervin Knuth, definisi algoritma adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan suatu masalah tertentu.

3. S. E. Goodman dan S.T. Hedetniemi Menurut Goodman dan Hedetniemi, pengertian algoritma adalah urutan terbatas dari operasi-operasi yang terdefinisi dengan baik, dimana masing-masing membutuhkan memori dan waktu yang terbatas untuk menyelesaikan suatu masalah.
4. Seymour Lipschutz dan Marc Lipson Menurut Seymour Lipschutz dan Marc Lipson (praktisi matematika dan komputer), pengertian algoritma adalah suatu daftar langkah demi langkah yang terhingga dari intruksi-intruksi yang terdefinisikan dengan jelas yang digunakan untuk memecahkan permasalahan tertentu.
5. Marvin Minsky Menurut Marvin Minsky (pakar Artificial Intelligence), pengertian algoritma adalah seperangkat aturan yang memberitahukan kepada kita dari waktu ke waktu, tepatnya bagaimana untuk bertindak.
6. Andrey Andreyevich Markov Menurut Andrey Andreyevich Markov (ahli matematika dari Rusia), pengertian algoritma adalah hal umum untuk dipahami sebagai suatu keputusan yang tepat untuk mendefinisikan proses komputasi yang mengarahkan dari data awal hingga hasil yang diinginkan.

Pada dasarnya tujuan dan fungsi utama dari algoritma adalah untuk memecahkan suatu masalah. Lebih jelasnya, adapun tujuan dan fungsi algoritma adalah sebagai berikut:

1. Untuk membantu menyederhanakan suatu program yang rumit dan besar.
2. Untuk memudahkan dalam membuat sebuah program untuk masalah tertentu.
3. Algoritma dapat digunakan berkali-kali untuk menyelesaikan suatu permasalahan.
4. Membantu memecahkan suatu permasalahan dengan logika dan sistematis.
5. Untuk meminimalisir penulisan program secara berulang-ulang.
6. Agar dapat melakukan pendekatan top-down dan divide and conquer.
7. Untuk memudahkan membuat program yang lebih rapih dan terstruktur sehingga lebih mudah dipahami dan dikembangkan.
8. Memudahkan proses modifikasi pada program karena bisa dilakukan hanya pada satu modul tanpa harus mengubah modul lainnya.
9. Ketika terjadi kesalahan, algoritma dapat membantu menemukannya karena alur kerja yang jelas.
10. Memudahkan proses dokumentasi.

Algoritma dapat diklasifikasikan berdasarkan implementasinya. Mengacu pada pengertian algoritma di atas, adapun klasifikasi algoritma adalah sebagai berikut:

1. Rekursi dan Iterasi; Algoritma rekursi adalah algoritma yang memanggil dirinya sendiri secara berulang-ulang. Sedangkan algoritma iterasi adalah algoritma yang memakai konstruktif berulang dimana terkadang terdapat data tambahan pada struktur yang dibuat.
2. Logical; Algoritma logical adalah algoritma yang dapat memposisikan diri seperti logika deduksi yang terkontrol.
3. Serial, Parallel, atau Terdistribusi; Algoritma serial adalah algoritma yang menjalankan satu instruksi saja. Algoritma parallel adalah algoritma yang dapat mengerjakan suatu perintah dalam waktu yang sama. Sedangkan algoritma terdistribusi adalah algoritma yang memakai banyak mesin yang terkoneksi dengan jaringan.
4. Deterministik atau Non-deterministik; Algoritma deterministik adalah algoritma yang dapat memecahkan suatu masalah dengan keputusan yang tepat. Sedangkan algoritma Non-deterministik adalah algoritma yang memecahkan suatu masalah dengan metode penerkaan.
5. Tepat atau Perkiraan; Suatu algoritma mungkin saja memiliki solusi yang tepat, atau setidaknya mempunyai perkiraan yang mendekati solusi yang benar. Dalam merumuskannya dapat dilakukan dengan strategi deterministic ataupun secara acak.
6. Algoritma Quantum; Algoritma quantum adalah algoritma yang menggunakan model realistik dari komputasi quantum.

1.8 LaTeX

Latex (*LaTeX*) adalah bahasa *markup* yang digunakan untuk membuat dokumen untuk pengetikan sistem *TeX*, yang dinamakan berdasarkan gaya penulisannya sebagai *LaTeX*[?]. Larex biasa disebut *word processor* (pengolahan kata, pembuat dokumen) yang mirip dengan *Microsoft Word* hanya saja *LaTeX* lebih cocok digunakan untuk membuat dokumen yang panjang. *LaTeX* digunakan karena kualitasnya yang tinggi dalam *typesetting* yang dapat dicapai oleh *TeX*. Sistem *typesetting* pada *LaTeX* menawarkan *Desktop Publishing* yang *Prorammable* batau dapat di tulis ulang dan fasilitas *ekstensi* untuk otomatisasi aspek-aspek dalam *typesetting* termasuk pemberian nomor, referensi silang, tabel dan bingkai, tata letak halaman, dan *bibliographies*.

1.9 SQL

Basis Data memiliki Bahasa khusus yang mengatur interaksi atau komunikasi antara pemakai dengan basis data di dalam mengelola atau mengorganisasikan data. Contoh bahasa khusus Basis data adalah (*StructureQueryLanguage*) sebuah bahasa yang digunakan untuk mengakses data dalam Basis Data Relasional. SQL secara de facto merupakan bahasa standar yang digunakan dalam RDMS (Relational Database Management System). SQL juga merupakan bahasa baku(ANSI/SQL) non prosedural dan berorientasi himpunan (set oriented language) SQL dapat digunakan baik secara interaktif atau ditempelkan pada sebuah program aplikasi. Standarisasi SQL dimulai pada tahun 1986 ditandai dengan dikeluarkannya standar SQL oleh ANSI,kemudian diperbaiki pada tahun 1989 kemudian diperbaiki kembali pada tahun 1992. Selain itu MySQL software merupakan suatu aplikasi yang sifatnya open source serta server basis data MySQL memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur client server atau embedded systems.Dikarenakan faktor open source dan popular tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar MySQL berkembang sudah mencapai versi 5. Beberapa sistem manejemen database relasional umum yang menggunakan SQL adalah: Oracle, Sybase, Microsoft SQL server, Acces, Ingres dan lain lain. Meskipun demikian perintah-perintah SQL standar seperti Select, Insert, Update, Delete, Create, Drop dapat digunakan pada seluruh DBMS tersebut. MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola database-database yang ada di dalamnya. Query adalah pertanyaan atau permintaan informasi tertentu dari sebuah basis data yang ditulis dalam format tertentu. Query juga dapat diidentifikasi sebagai perintah yang digunakan untuk mengakses data pada sistem Basis Data untuk melakukan manipulasi terhadap Basis Data yang dikenal dengan nama SQL (Structured Query Language). Secara umum SQL terdiri dari dua bahasa, yaitu:

Beberapa jenis perintah SQL yang perlu diketahui Setelah kamu mengetahui sejarah dan pengertian SQL, kini saatnya untuk lebih mengetahui apa saja perintah-perintah dasar yang terdapat dalam SQR tersebut. Setidaknya ada tiga jenis perintah dasar dalam SQL yang penjabarannya akan diulas dalam poin-poin berikut ini.

1.9.1 Data Definition Language (DDL)

Jenis perintah dasar yang pertama adalah Data Definition Language atau biasa disingkat dengan DDL. Perintah dasar ini sebenarnya merupakan perintah paling mendasar dari bahasa SQL. Tujuannya untuk membuat struktur sebuah database. Kemudian, perintah dasar DDL masih dibedakan lagi ke dalam setidaknya lima jenis perintah yakni bisa kamu lihat di bawah ini.

1. Perintah Create: sebuah perintah yang bisa kamu gunakan ketika membuat sebuah database yang baru, baik itu berupa tabel baru atau sebuah kolom baru.

Kamu bisa membuat sebuah query dengan contoh CREATE DATABASE nama database.

2. Perintah Alter: biasa digunakan ketika seseorang ingin mengubah struktur tabel yang sebelumnya sudah ada. Bisa jadi dalam hal ini adalah seperti nama tabel, penambahan kolom, mengubah, maupun menghapus kolom serta menambahkan atribut lainnya.
3. Perintah Rename: dapat kamu gunakan untuk mengubah sebuah nama di sebuah tabel ataupun kolom yang ada. Bila kamu menggunakan perintah ini maka query-nya menjadi RENAME TABLE nama tabel lama TO nama tabel baru
4. Perintah Drop: Bisa kamu gunakan dalam menghapus baik itu berupa database, table maupun kolom hingga index.
5. Perintah Show: perintah DDL ini digunakan untuk menampilkan sebuah tabel yang ada.

1.9.2 Data Manipulation Language (DML)

Berikutnya dalam pengertian SQL, dikenal dengan adanya Data Manipulation Language (DML). Seperti namanya, perintah dasar SQL ini bertujuan untuk memanipulasi data yang ada dalam sebuah database. Perintah dalam DML juga terbagi ke dalam empat jenis. Beberapa di antaranya adalah insert, select, update, dan delete.

1. Perintah Insert: Kamu bisa menggunakan perintah ini untuk memasukkan sebuah record baru di dalam sebuah tabel database.
2. Perintah Select: Pada perintah ini kamu dapat menggunakanya dalam menampilkan maupun mengambil sebuah data pada tabel. Data yang diambil pun tidak hanya terbatas pada satu jenis saja melainkan lebih dari satu tabel dengan memakai relasi.
3. Perintah update: Ini dapat kamu gunakan ketika ingin melakukan pembaruan data di sebuah tabel. Contohnya saja jika ada kesalahan ketika memasukkan sebuah record. Kamu tidak perlu menghapusnya dan bisa diperbaiki menggunakan perintah ini.
4. Perintah Delete: Perintah DML ini dapat digunakan ketika kamu ingin menghapus sebuah record yang ada dalam sebuah tabel.

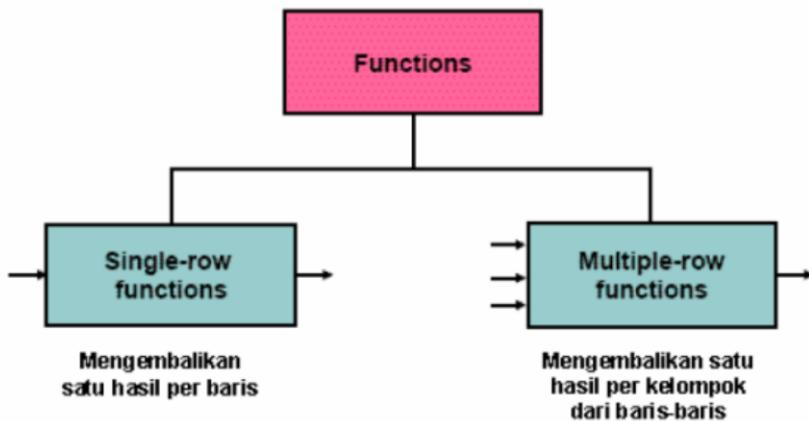
1.9.3 Data Control Language (DCL)

Perintah dasar berikutnya adalah Data Control Language atau DCL. Perintah SQL ini digunakan khususnya untuk mengatur hak apa saja yang dimiliki oleh pengguna. Baik itu hak terhadap sebuah database ataupun pada tabel maupun field yang ada. Melalui perintah ini, seorang admin database bisa menjaga kerahasiaan sebuah

database. Terutama untuk yang penting. DCL berdasarkan perintah dasarnya terbagi dalam dua perintah utama yakni:

1. Perintah Grant: Perintah ini biasanya digunakan ketika admin database ingin memberikan hak akses ke user lainnya. Tentu pemberian hak akses ini dapat dibatasi atau diatur. Dalam hal ini admin pun dapat memberikan akses mengenai perintah dalam DML di atas.
2. Perintah Revoke: Kebalikannya dari Grant, Revoke terkadang sering digunakan untuk mencabut maupun menghapus hak akses seorang pengguna yang awalnya diberikan akses oleh admin database melalui perintah Grant sebelumnya.

1.9.4 Fungsi SQL



Gambar 1.1 Fungsi SQL

1.9.4.1 SingleRow Functions Fungsi-fungsi ini hanya digunakan pada baris-baris tunggal dan mengembalikan satu hasil per baris.

Ada tipe-tipe berbeda dari singlerow functions. Pelajaran ini mencakup hal-hal berikut :

1. Character (karakter) : Menerima input karakter dan dapat mengembalikan baik nilai-nilai karakter ataupun angka.
2. Number (angka) : Menerima masukkan angka dan mengembalikan nilai-nilai angka
3. Date (tanggal) : Bekerja pada nilai-nilai dari tipe data DATE (semua date functions mengembalikan suatu nilai dari tipe data DATE kecuali fungsi MONTHS BETWEEN, yang mengembalikan suatu angka.)
4. Conversion (konversi) : Mengkonversi suatu nilai dari satu tipe data ke tipe data lainnya.
5. General (umum) : (fungsi-fungsi umum) :
 - (a) NVL
 - (b) NVL2
 - (c) NULLIF
 - (d) COALESCE
 - (e) CASE
 - (f) DECODE
 - (g) Single Rows Functions

Singlerow function digunakan untuk memanipulasi itemi-temdata. Singlerow functions menerima satu atau lebih argumen dan mengembalikan satu nilai untuk setiap baris yang dihasilkan oleh suatu query. Suatu argumen dapat berupa berikut ini :

- (a) Usersupplied
- (b) constant (Konstanta yang disediakan oleh user)
- (c) Nilai variable
- (d) Nama Kolom
- (e) Ekspresi

Fitur-fitur dari singlerow functions mencakup :

- (a) Aksi pada setiap baris yang dikembalikan di dalam query
- (b) Mengembalikan satu hasil per baris

- (c) Memungkinkan pengembalian suatu nilai data dari suatu tipe berbeda dari pada satu yang direferensikan.
- (d) Mungkin menerima satu atau lebih argumentargumen
- (e) Dapat digunakan didalam klausa klausa

SELECT, WHERE, dan ORDER BY; dapat disarangkan (nested)

Character Functions

Single-row character functions menerima data karakter sebagai masukan dan dapat mengembalikan baik nilai-nilai karakter maupun angka. *Character functions* dapat dibagi menjadi :

- *Case-manipulation functions*
- *Character-manipulation functions*

Function	Melakukan
<code>LOWER(column/expression)</code>	Mengkonversi nilai-nilai karakter huruf menjadi <i>lowercase</i> (huruf kecil)
<code>UPPER(column/expression)</code>	Mengkonversi nilai-nilai karakter huruf menjadi <i>uppercase</i> (huruf besar)
<code>INITCAP(column/expression)</code>	Mengkonversi nilai-nilai alpha karakter menjadi <i>uppercase</i> untuk huruf pertama dari tiap kata; semua huruf-huruf lain <i>lowercase</i>
<code>CONCAT(column1/expression1, column2/expression2)</code>	Menggabungkan nilai karakter pertama ke karakter kedua : sama dengan operator penggabungan ()
<code>SUBSTR(column/expression, m[,n])</code>	Menghasilkan karakter-karakter tertentu dari nilai karakter dimulai pada posisi karakter <i>ke-m</i> , <i>ke-n</i> panjang karakter (jika <i>m</i> adalah negatif, dihitung mulai dari akhir nilai karakter. Jika <i>n</i> dihilangkan, menghasilkan semua karakter sampai akhir dari rangkaian.)

Catatan : Fungsi-fungsi yang didiskusikan dalam pelajaran ini adalah hanya beberapa dari fungsi-fungsi yang ada.

Gambar 1.2 Fungsi pada character

Fungsi-Fungsi Character (lanjutan)

Function	Melakukan
LENGTH (<i>column/expression</i>)	Mengembalikan jumlah karakter dalam ekspresi
INSTR (<i>column/expression</i> , 'string', [<i>m</i>], [<i>n</i>])	Mengembalikan posisi numerik dari suatu rangkaian penamaan. Secara <i>optional</i> , Anda dapat menyediakan suatu posisi ke- <i>m</i> untuk memulai pencarian, dan yang terjadi di- <i>n</i> dari suatu rangkaian. <i>m</i> dan <i>n</i> <i>default</i> -nya 1, artinya pencarian dimulai di awal suatu pencarian dan melaporkan kejadian yang pertama.
LPAD (<i>column /expression</i> , <i>n</i> , 'string')	Mengisi nilai karakter perataan kanan (<i>right-justified</i>) ke suatu lebar total <i>n</i> posisi karakter
RPAD (<i>column /expression</i> , <i>n</i> , 'string')	Mengisi nilai karakter perataan kiri (<i>left-justified</i>) ke suatu lebar total <i>n</i> posisi karakter
TRIM (<i>leading/trailing/both</i> , <i>trim_character</i> FROM <i>trim_source</i>)	Memungkinkan Anda untuk memotong karakter-karakter bagian awal atau bagian akhir atau kedua-duanya dari suatu rangkaian karakter. Jika <i>trim_character</i> atau <i>trim_source</i> adalah suatu karakter literal, Anda harus mengapitnya dengan tanda petik tunggal. Ini adalah fitur yang ada di Oracle8i dan versi selanjutnya.
REPLACE (<i>text</i> , <i>search_string</i> , <i>replacement_string</i>)	Mencari suatu ekspresi teks untuk suatu rangkaian karakter dan, jika ditemukan, digantikan dengan rangkaian yang telah ditentukan.

Gambar 1.3 Fungsi pada character

1.9.4.2 Multiplerow Functions Fungsi-fungsi dapat memanipulasi kelompok dari baris-baris untuk memberi suatu hasil baris-baris per kelompok. Fungsi-fungsi ini dikenal juga sebagai group functions

(a) CaseManipulation Functions

- i. LOWER : Konversi mixedcase (bentuk campuran) atau karakter karakter string huruf besar ke huruf kecil
- ii. UPPER : Konversi mixedcase atau karakter karakter string huruf kecil ke huruf besar
- iii. INITCAP : Konversi huruf pertama setiap kata ke huruf besar dan huruf-huruf selanjutnya ke huruf kecil

```
SELECT the job id for ——UPPER(last name)—— is ——LOWER(job
id) AS EMPLOYEE DETAILS
FROM employees;
```

Menggunakan Fungsi-Fungsi Case Manipulation Klausa WHERE dapat ditulis ulang dengan cara berikut untuk menghasilkan hasil yang sama :

. . . WHERE last name= Higgins Nama pada output muncul seperti nama itu disimpan di dalam database. Untuk menampilkan nama dimana hanya huruf pertama dalam bentuk huruf besar, gunakan fungsi UPPER pada pernyataan SELECT.

```
SELECT employee id, UPPER (last name), department id
FROM employees
WHERE INITCAP(last name) = Higgins;
```

Fungsi-Fungsi Number

Fungsi-fungsi *Number* menerima input numerik dan mengembalikan nilai-nilai numerik. Pada bagian ini dibahas beberapa fungsi-fungsi *number*.

Function	Purpose
ROUND (<i>column</i> <i>expression</i> , <i>n</i>)	Membulatkan kolom, ekspresi, atau nilai posisi ke- <i>n</i> desimal atau, jika <i>n</i> dihilangkan, tidak ada posisi desimal (Jika <i>n</i> adalah negatif, angka-angka di kiri dari desimal dibulatkan.)
TRUNC (<i>column</i> <i>expression</i> , <i>n</i>)	Memotong kolom, ekspresi, atau nilai posisi ke- <i>n</i> desimal atau, jika <i>n</i> dihilangkan, <i>n default</i> -nya nol
MOD (<i>m</i> , <i>n</i>)	Mengembalikan sisa dari <i>m</i> yang dibagi oleh <i>n</i>

Catatan : Daftar ini hanya berisi beberapa dari *number function* yang tersedia . Untuk informasi lebih lanjut, lihat "Number Functions" di *Oracle SQL Reference*.

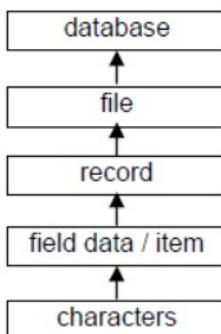
Gambar 1.4 Fungsi number

- (a) CONCAT : Menggabungkan bersama nilai-nilai (Anda dibatasi untuk menggunakan dua parameter pada CONCAT.)
- (b) SUBSTR : Memotong suatu rangkaian dari panjang tertentu
- (c) LENGTH : Menampilkan panjang dari suatu rangkaian sebagai suatu nilai numeric
- (d) INSTR : Menemukan posisi numerik dari suatu karakter nama
- (e) LPAD : Mengisi nilai karakter rightjustified (perataan kanan)
- (f) RPAD : Mengisi nilai karakter leftjustified(perataan kiri)
- (g) TRIM : Memotong karakter-karakter bagian awal atau bagian akhir (atau keduaduanya) dari suatu rangkaian karakter (jika trim character atau trim source adalah suatu karakter literal, Anda harus mengapitnya didalam tanda petik tunggal.)

BAB 2

HIRARKI JENJANG DATA

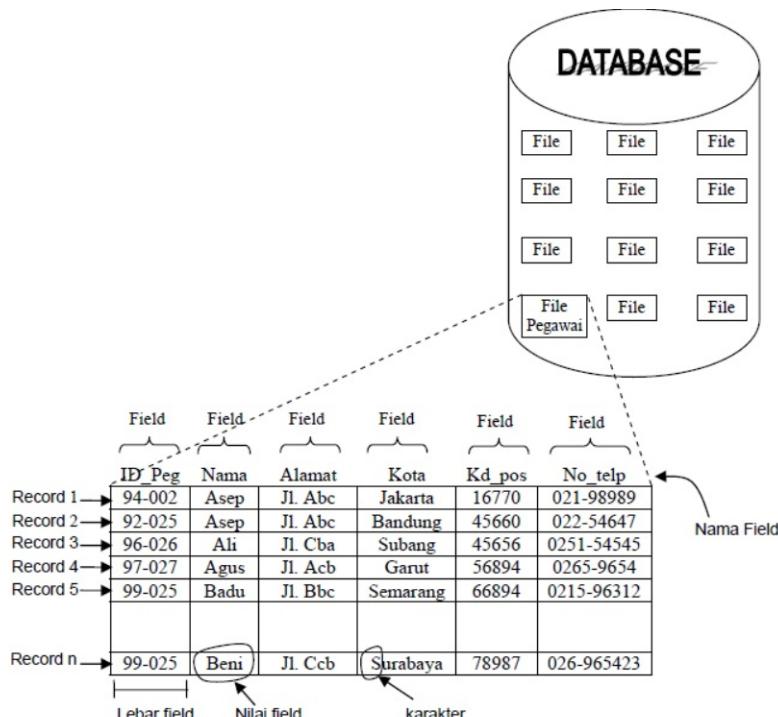
2.1 Hirarki / Jenjang Data



Gambar 2.1 Hierarki/ Jenjang Data

1. Characters : merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (special characters) yang membentuk suatu item data / field.
2. Field : merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record.
3. Record : Kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap-tiap record dapat mewakili data tiap-tiap karyawan.
4. File: File terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis. Misalnya file mata pelajaran berisi data tentang semua mata pelajaran yang ada.
5. Database : Kumpulan dari file / tabel membentuk suatu database

Gambar berikut adalah contoh tentang jenjang data:



Gambar 2.2 Basis Data

Selanjutnya contoh tabel mahasiswa:

The diagram shows a table with 5 rows and 3 columns. The columns are labeled 'NOBP', 'NamaMahasiswa', and 'Alamat'. The rows contain student data. Below the table, arrows point from labels to specific cells: 'Field Nop' to '03156001', 'Kolom Field Nama Mahasiswa' to 'Helga Elzar Adha', and 'Kolom field Alamat' to 'Curup'. To the left of the table, an arrow points from 'Data Value' to the first row.

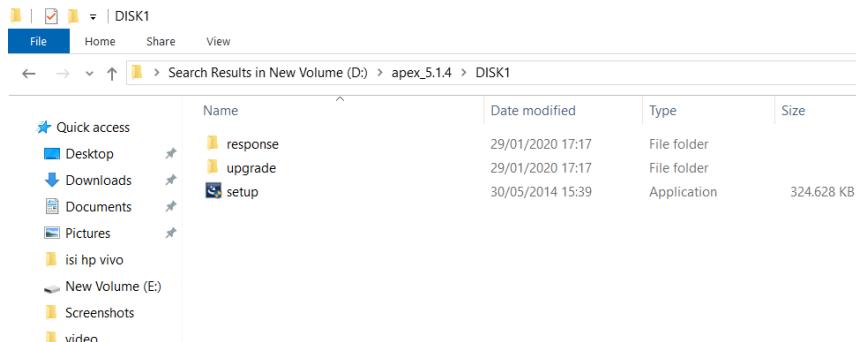
NOBP	NamaMahasiswa	Alamat
03156001	Helga Elzar Adha	Curup
03156002	Rifky Zarel Putra	Bengkulu
03156003	Fikri Putra Zarel	Padang
03156004	Reinhard Steven	Padang Panjang
03156005	Edi Brocoli	Lubuk Alung
Field Nop	Kolom Field Nama Mahasiswa	Kolom field Alamat

Gambar 2.3 Basis Data

2.2 Tahapan Instalasi Oracle Apex Offline

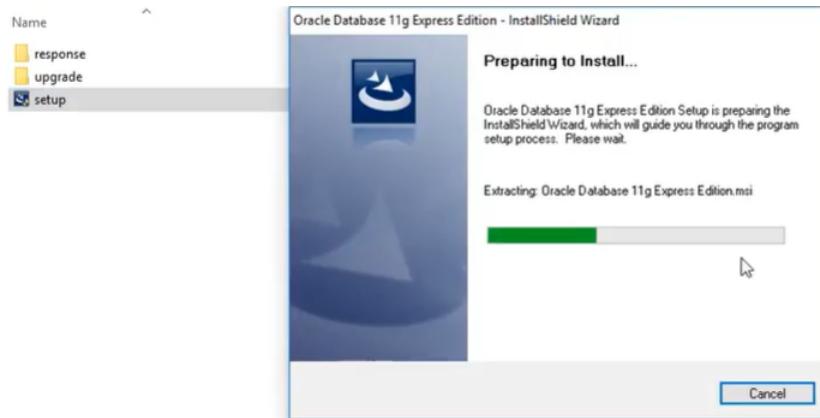
Oracle apex memiliki berbagai macam jenis dalam penggunaanya. *Oracle apex* memiliki 2 macam yaitu, *oracle apex online* dan *oracle apex offline*. Berikut ini adalah langkah-langkah untuk melakukan instalasi *oracle apex offline*

1. Pertama buka folder apex yang sudah tersedia di dalam file komputer / laptop



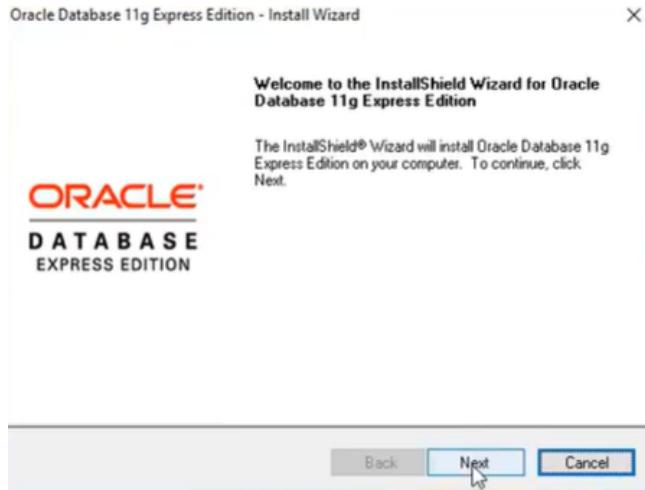
Gambar 2.4 File apex yang ada di Pc

2. Langkah kedua yaitu adalah melakukan proses *running* pada program setup seperti gambar dibawah ini



Gambar 2.5 Proses Penginstallan Aplikasi

3. Apabila proses seperti diatas telah selesai maka klik *next* pada tampilan seperti berikut



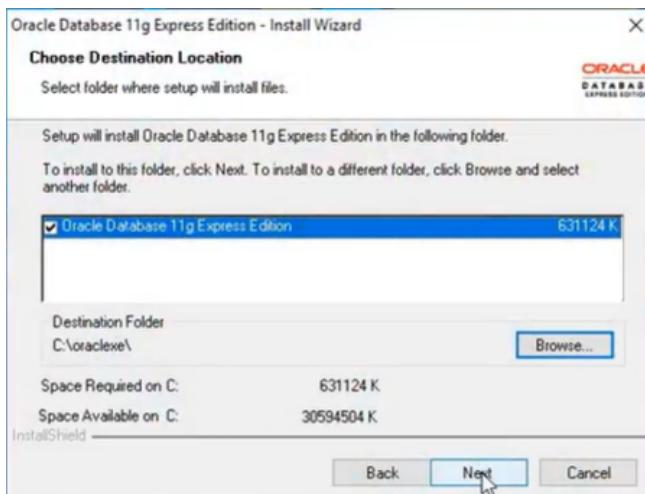
Gambar 2.6 Proses Penginstallan Aplikasi

4. Pada proses ini adapun serangkaian perintah yang harus dilakukan, salah satunya pada gambar berikut. Untuk melakukannya user harus memilih i accept the terms in the license agreement yang bertujuan bahwa user menerima ketentuan dalam perjanjian lisensi. Lalu klik next untuk melanjutkan proses selanjutnya



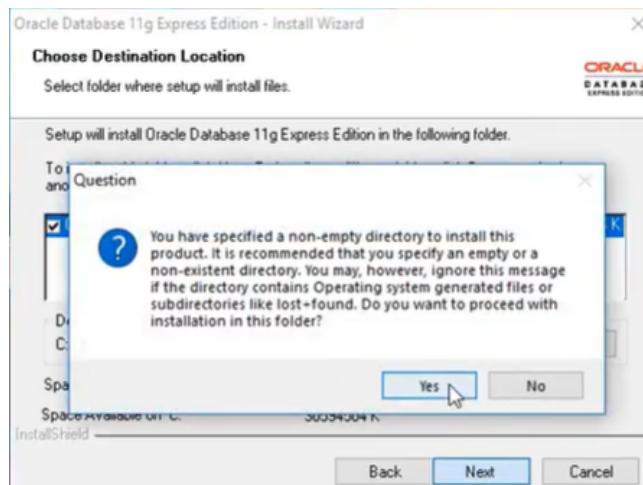
Gambar 2.7 Proses Penginstallan Aplikasi

5. Langkah selanjutnya adalah dengan cara klik "oracle database 11g Express Edition" dengan begitu semua file dalam pengaksesan masuk ke dalam folder C: , klik next untuk mengikuti proses selanjutnya



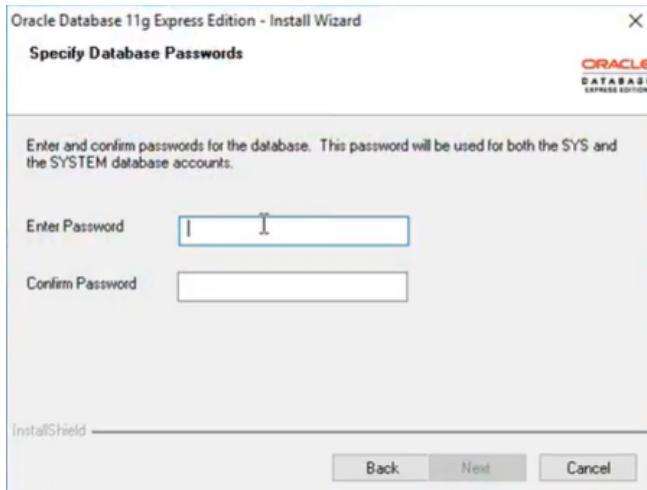
Gambar 2.8 Proses Penempatan File Aplikasi

6. Klik **yes** untuk menyetujui bahwa user telah menentukan direktori tidak kosong untuk menginstal produk ini. dan disarankan agar user menentukan direktori kosong atau tidak ada. user dapat, mengabaikan pesan dibawah ini apabila direktori berisi file atau subdirektori yang dihasilkan sistem operasi seperti hilang ditemukan.Dan klik **yes** untuk melanjutkan installasi.+



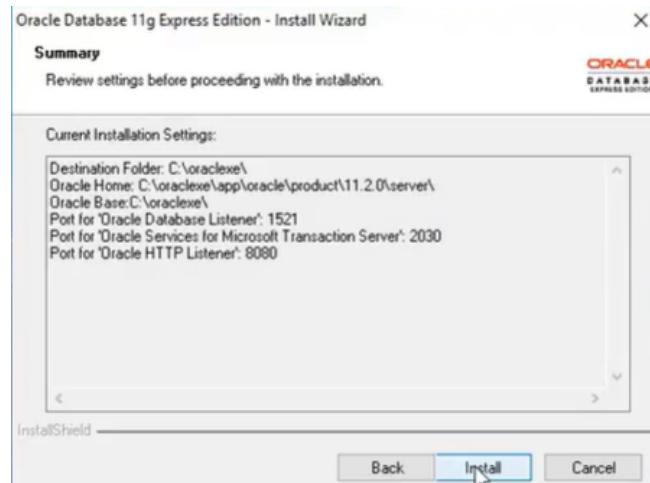
Gambar 2.9 Proses Penempatan File Aplikasi

7. Spesifikasi untuk pengisian *password*. Password harus berisikan alfabet, angka dan karakter lalu tekan *next*



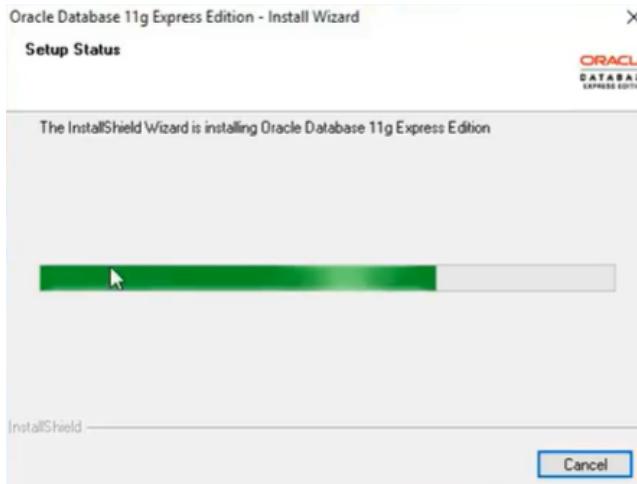
Gambar 2.10 Proses Pembuatan password Aplikasi

8. Tahapan selanjutnya adalah *review* tentang apa yang telah dibuat pada proses yang telah ada. Apabila dalam *setting pengreviewan* sudah sesuai maka klik *install*. Apabila belum sesuai maka tekan *back* untuk kembali ke tahapan sebelumnya.



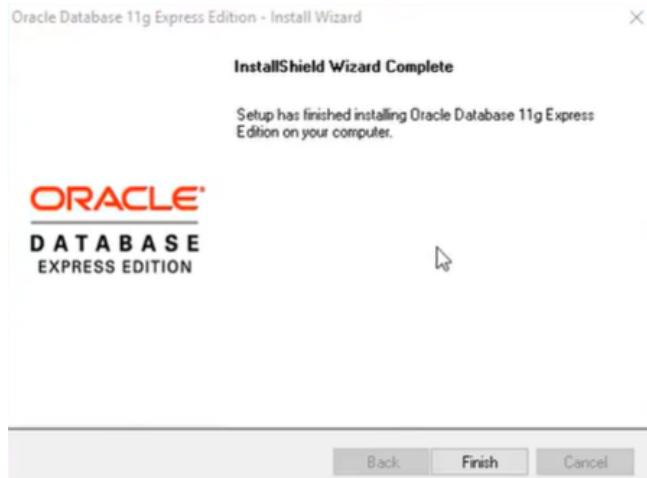
Gambar 2.11 Review Instalasi pada Aplikasi

9. Tunggu untuk beberapa saat dalam proses penginstalan seperti pada gambar berikut. Dalam proses tersebut membutuhkan *loading* sekitar 1 sampai 2 menit.



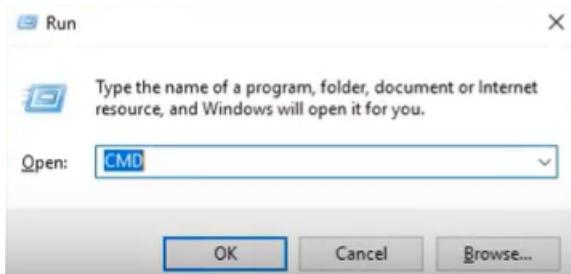
Gambar 2.12 Installasi Aplikasi

10. Seperti pada gambar dibawah ini. Installasi *Oracle Apex Offline* sudah terinstall. Namun masih ada langkah selanjutnya. klik *finish* untuk selesai pada menyelesaikan proses ini.



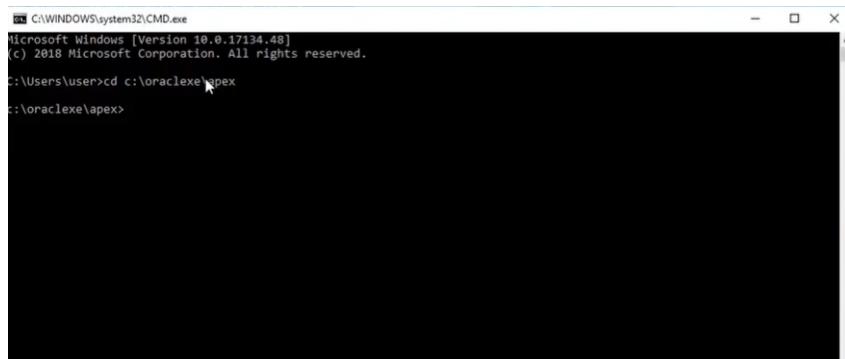
Gambar 2.13 Installasi Aplikasi Telah Berhasil

11. Tahapan selanjutnya adalah buka cmd dengan cara tekan tombol pada *keyboard* yaitu *windows + R*.lalu klik ok untuk melanjutkan tahapan selanjutnya, dan klik cancle untuk mehentikan proses ini.



Gambar 2.14 CMD

12. Apabila sudah membuka *CMD* maka ketik *cd c: .* cd adalah perintah dimana kita akan masuk ke dalam suatu folder tertentu, *oracleexe* merupakan file dimana user menyimpan installasi apex yang telah di install. lalu tekan enter.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.48]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user>cd c:\oracleexe\apex
C:\oracleexe\apex>
```

Gambar 2.15 CMD

13. Ketik *SQL Plus* lalu enter. Maka akan muncul kolom untuk mengisikan *user-name*. Pada kolom tersebut tulis *System/abc as sysdba* tekan enter untuk mengkoneksikan ke *database 11g express relase 11.2.0.2.0 - 64bit production*. Tekan enter

```
c:\oracle\product\11.2.0\client_1\bin>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on Jum Jun 1 23:30:31 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: system/abc123 as sysdba

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> -
```

Gambar 2.16 Pengkonfigurasian pada CMD

14. Selanjutnya ketik `@apexins.sql sysaux sysaux temp /i/` lalu enter maka akan terjadi proses perngkonfigurasian

```
SQL> @apexins.sql sysaux sysaux temp /i/  
-
```

Gambar 2.17 Pengkonfigurasian pada CMD

15. Pada proses pengkonfigurasian seperti pada gambar dibawah ini membutuhkan waktu yang lumayan lama sekitar 5-10 menit tergantung kekuatan masing-masing PC.

```
...Installing Theme 2

old  5:      wwv_flow_application_install.set_offset(^THEME_OFFSET);
new  5:      wwv_flow_application_install.set_offset(1447581811918206904);
API Last Extended:20160824
Your Current Version:20160824
This import is compatible with version: 20160824
COMPATIBLE (You should be able to run this import without issues.)
--application/set_environment
APPLICATION 108802 - Theme: Builder Blue
--application/delete_application
--application/ui_types
--application/create_application
--application/shared_components/navigation/lists
--application/shared_components/files
--application/plugin_settings
--application/shared_components/security/authorizations
--application/shared_components/navigation/navigation_bar
--application/shared_components/logic/application_processes
--application/shared_components/logic/application_items
--application/shared_components/logic/application_computations
--application/shared_components/navigation/tabs/standard
--application/shared_components/navigation/tabs/parent
--application/shared_components/user_interface/lovs
--application/shared_components/navigation/trees
--application/pages/page_groups
--application/shared_components/navigation/breadcrumbs/main_menu
--application/shared_components/user_interface/templates/page
```

Gambar 2.18 Pengkonfigurasian

16. Apabila dalam penkonfigurasian telah berelasi dengan oracle database maka ketik *Sql plus* lalu enter. Setelah itu isikan user name *syste./abc123 as sysdba* lalu enter maka akan terkoneksi dengan *oracle database 11g express edition relase 11.2.0.2.0 - 64bit production* seperti proses yang telah dilakukan diatas dan diulangi kembali.

```
c:\oraclexe\apex>sqlplus  
SQL*Plus: Release 11.2.0.2.0 Production on Jum Jun 1 23:41:15 2018  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Enter user-name: system/abc123 as sysdba  
Connected to:  
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

Gambar 2.19 Pengkonfigurasian ke Database

17. ketik *alter user anonymos account unlock;* lalu enter. Proses ini berfungsi untuk membuka user anonymos

```
SQL> alter user anonymous account unlock;  
User altered.
```

Gambar 2.20 Membuka User Anonymos

18. Langkah selanjutnya adalah ketik `@apxconf.sql` lalu tekan enter. Maka akan muncul port 8080 dan muncul administrator username [admin]. Tekan enter pada user admin dan email admin setelah itu masukkan password yang telah dibuat seperti yang sudah dibuat pada proses diatas yang mengandung alfabet, angka dan karakter

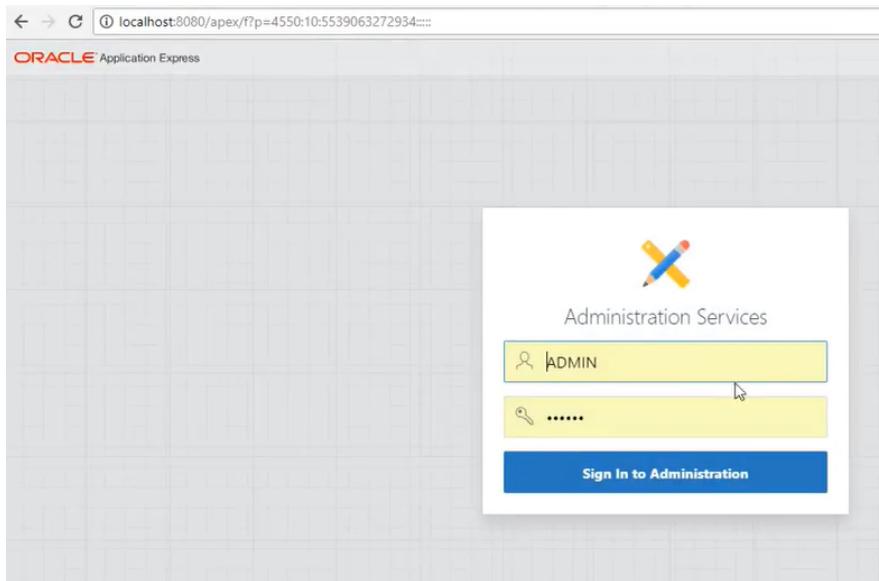
```
SQL> @apxconf.sql
PORT
-----
8080

Enter values below for the XDB HTTP listener port and the password for the Application Express ADMIN user.
Default values are in brackets [ ].
Press Enter to accept the default value.

=====
This script can be used to change the password of an Application Express
instance administrator. If the user does not yet exist, a user record will be
created.
=====
Enter the administrator's username [ADMIN]
User "ADMIN" exists.
Enter ADMIN's email [ADMIN]
Enter ADMIN's password []
Changed password of instance administrator ADMIN.
```

Gambar 2.21 Administrator pada aplikasi

19. Langkah terakhir adalah buka *google* lalu buka link *localhost/8080/apex/apex_admin*.



Gambar 2.22 Administrator pada aplikasi

2.3 Pembuatan Tabel Pada SQL Command

2.3.1 To create a new table Using SQL Comand :

- Pada Database Home Page, klik SQL icon, kemudian pilih SQL COMMANDS
- Tulis SQL syntax. Contoh sintax untuk membuat table mhs :

```
CREATE TABLE mhs(
nim varchar(12) NOT NULL,
nama varchar(50) NOT NULL,
sex   CHAR NOT NULL
      CHECK (sex IN ('M', 'F')),
alamat varchar(50) DEFAULT NULL,
jurusan NOT NULL,
PRIMARY KEY(nim),
);
```

Gambar 2.23 Contoh Syntax Sql

- Kemudian klik Run
- Dan table di buat.

2.3.2 Show description Table :

```
DESC table_name;
```

Gambar 2.24 Diskripsi pada tabel

2.3.3 Delete table Using SQL Comand :

```
DROP TABLE      table_name;
```

Gambar 2.25 Delete Pada Tabel

2.3.4 Change table structure Using SQL Comand :

```
ALTER TABLE table_name alter_option;
```

Gambar 2.26 Alter Table

2.3.5 Delete table Using SQL Comand :

```
ALTER TABLE table_name alter_option;
```

Gambar 2.27 Alter Table

Alter option adalah perintah sepesifik untuk mengubah struktur table :

- (a) ADD new field;
- (b) ADD INDEX index name;
- (c) ADD PRIMARY KEY(Key Field)
- (d) CHANGE Field target New defintion Field
- (e) MODIFY Field definition
- (f) DROP Field name to delete;
- (g) RENAME TO New Table Name;

berikut contohnya:

```
ALTER TABLE mhs ADD religi VARCHAR(15) NOT NULL;
ALTER TABLE mhs ADD PRIMARY KEY(nim);
ALTER TABLE mhs CHANGE religi religi VARCHAR(2) NOT NULL;
ALTER TABLE mhs MODIFY religi CHAR(2) NOT NULL;
ALTER TABLE mhs DROP religi;
ALTER TABLE mhs RENAME TO mahasiswa;
```

Gambar 2.28 contoh

2.3.6 Insert, Update, and Delete Record dengan SQL Command

2.3.6.1 Insert Record Untuk memasukan nilai atau insert record ke table teradapat tiga cara :

```
INSERT INTO table_name VALUES ('value 1', 'value 2', '....');
```

```
INSERT INTO table_name (field 1, field 2, field 3, ....) VALUES ('value 1', 'value 2',
```

```
INSERT INTO table_name SET (field 1 ='value 1', field 2= 'value 2', '....');
```

Gambar 2.29 Insert Record

```
INSERT INTO mhs VALUES('41507010080','JOKO','M', 'Jakarta','');
```

Gambar 2.30 contoh

2.3.6.2 Update Record Untuk mengupdate nilai record dapat di lakukan dengan perintah berikut :

```
UPDATE table_name SET Field 1 = 'New_Value 1', Field 2 = 'New_Value 2',  
Field .... = 'New_Value ....' [WHERE CONDITION_FIELD];
```

Gambar 2.31 update

```
UPDATE mhs SET alamat='Bogor', sex='F' WHERE nim='41507010069';
```

Gambar 2.32 Contoh update

2.3.7 Delete Record

Untuk menghapus nilai dari suatu record dapat di lakukan dengan perintah berikut :

```
DELETE FROM table_name [WHERE CONDITION_FIELD];
```

Gambar 2.33 Delete

```
DELETE FROM mhs WHERE nim='41507010069';
```

Gambar 2.34 Contoh Delete

2.3.8 Membuat Table dengan Primer dan Komposit Key

(a) Primary Key :

Primary key dari suatu tabel harus berisi nilai yang unik, dan non-null untuk setiap barisnya

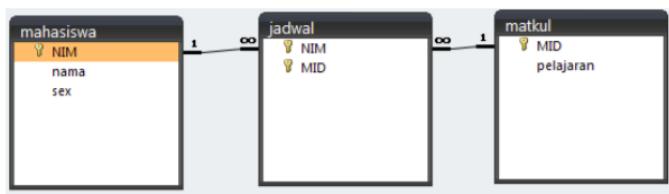
(b) Foreign Key

Foreign Key : adalah kolom atau himpunan kolom yang menghubungkan setiap baris dalam child table yang berisi Foreign Key dengan baris dari parent table yang berisi Primary Key yang sesuai/match.

(c) Candidate Key

- Superkey (K) dalam relasi

- Setiap relasi R, nilai K akan mengidentifikasi secara unik tuplenya. Jika Candidate key terdiri dari beberapa atribut, disebut composite key.



Gambar 2.35 contoh perhatikanlah Skema Kampus

Skema tersebut menggambarkan setiap mahasiswa dapat mengambil banyak mata kuliah, dan setiap mata kuliah dapat di ambil oleh banyak mahasiswa.

Untuk membuat foreign key serta cara membuat composite key pada table jadwal dapat menggunakan perintah SQL berikut :

Table Mahasiswa

```
CREATE TABLE "MAHASISWA" (
    "NIM" VARCHAR2(12) NOT NULL ENABLE,
    "NAMA" VARCHAR2(50) NOT NULL ENABLE,
    "SEX" CHAR(1) NOT NULL ENABLE,
    CHECK (sex IN ('M', 'F')) ENABLE,
    PRIMARY KEY ("NIM") ENABLE );
```

Gambar 2.36 Contoh Tabel Mahasiswa

Table Matkul

```
CREATE TABLE "MATKUL"
(
    "MID" VARCHAR2(12) NOT NULL ENABLE,
    "PELAJARAN" VARCHAR2(50),
    PRIMARY KEY ("MID") ENABLE );
```

Gambar 2.37 Contoh Tabel Matkul

```
CREATE TABLE jadwal(
nim varchar(12) NOT NULL,
mid varchar(12) NOT NULL,
FOREIGN KEY (mid) REFERENCES matkul , FOREIGN KEY (nim) REFERENCES mahasiswa
PRIMARY KEY("NIM","MID") );
```

Gambar 2.38 Tabel Jadwal

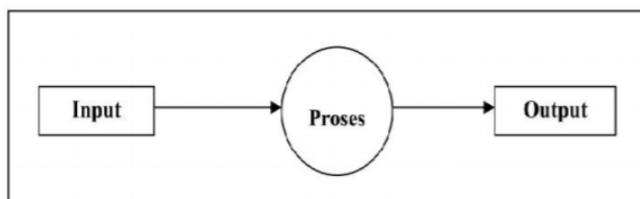
Tipe Data	Penjelasan
char(n)	String sepanjang n karakter. Bila n tidak disertakan, maka panjang karakter adalah 1. Contoh : $\text{char} = 1, \text{char}(3) = 3$
varchar(n)	String yang panjangnya bisa berubah-ubah sesuai kebutuhan, namun string tersebut dibatasi sebanyak n karakter. (<i>Oracle merekomendasikan varchar2. Bila diketikan varchar pada oracle maka otomatis berubah menjadi varchar2</i>). Contoh : $\text{varchar}(5) = \text{batas varchar adalah } 5$
varchar2(n)	String yang panjangnya bisa berubah –ubah sesuai dengan kebutuhan, namun string tersebut dibatasi sebanyak n karakter. Maksimum karakter pada varchar2 adalah 2000 karakter. Contoh : $\text{varchar2}(10) = \text{batas varchar2 adalah } 10$
long	Tipe data binary, maksimum 2 Giga Byte, disimpan dalam format internal Oracle.
long raw	Sama dengan long yaitu tipe data binary, maksimum 2 Giga Byte, tidak dikonversi oleh Oracle (data mentah apa adanya).
date	Tanggal, menyimpan tahun, bulan, hari, jam, menit, dan detik.
number(n,p)	Angka pecahan, baik fixed decimal atupun floating point. Nilai n adalah jumlah bytes total dan p adalah presisi angka di belakang koma. Contoh : $\text{number}(4,2) = 4 \text{ (jumlah batas), } 2 \text{ (presisi angka di belakang koma)}$

Gambar 2.39 Tabel Tipe Data

2.4 Perancangan sistem

Perancangan sistem adalah merancang atau mendesain suatu system yang baik yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan para pemakai sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalam. Selain itu Suatu sistem terdiri dari sistem-sistem bagian (subsystems). Masing-masing subsistem terdiri dari subsistem-subsystem yang lebih kecil lagi atau terdiri dari komponen-komponen. Interaksi dari subsistem-subsystem sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (integrated). Keterpaduan sistem ini memungkinkan terciptanya kerjasama untuk menghasilkan informasi yang cepat, tepat dan akurat. Pengertian sistem menurut Jogiyanto (2005 : 1) yang lebih menekankan pada prosedurnya didefinisikan sebagai berikut : Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urut-urutan operasi di dalam sistem. Prosedur (procedure) didefinisikan oleh Richard F. Neuschel yang disadur oleh Jogiyanto (2005 : 1) mendefinisikan sebagai berikut : Prosedur adalah suatu urut-urutan operasi klerikal (tulis-menulis) biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis terjadi. Lebih lanjut Jerry FitzGerald, Adra F.FitzGerald dan Warren D.Stalling, Jr., mendefinisikan prosedur sebagai berikut: Suatu prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (what) yang harus dikerjakan, siapa (who) yang mengerjakannya, kapan (when) dikerjakan dan bagaimana (how) mengerjakannya. Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut ini: Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencari suatu tujuan tertentu.

Bentuk umum sistem dari suatu sistem terdiri atas masukan (Input), proses dan keluaran (Output), dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan diproses dan akan menghasilkan suatu keluaran.



Gambar 2.40 Bentuk Umum Sistem

Elemen Sistem, Semua sistem meliputi tiga elemen utama yaitu input, transformasi dan output. Sebagian sistem dapat mengendalikan operasi mereka sendiri yang disebut sebagai sistem lingkaran tertutup (closed-loop system). Sistem lingkaran tertutup mencakup suatu mekanisme kontrol, tujuan dan lingkaran umpan balik (feed-back loop) disamping tiga elemen utama. Sistem yang tidak memiliki kemampuan pengendalian disebut sistem lingkaran terbuka (open-loop system), dalam arti mereka berhubungan dengan lingkungan mereka. Perusahaan adalah suatu contoh sistem terbuka dan sistem lingkaran tertutup.

2.4.1 Macam - Macam Karakteristik Sistem

Menurut Jogiyanto (2005:3) Sistem memiliki karakteristik atau sifat-sifat yang tertentu yaitu mempunyai komponen (Components), batas sistem (Boundary), lingkungan (Environments), penghubung (Interface), masukan (Input), keluaran (Output), pengolah (Process), dan sasaran (Objectives) atau tujuan (Goals).12 Menurut Jogiyanto (2005:3-5) Sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan.

2. Sifat-sifat

Sistem itu terdiri untuk menjalankan fungsi tertentu dan mempunyai sistem yang lain secara keseluruhan.

3. Batasan (Boundary) Sistem

Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.

4. Lingkungan Luar Sistem (Environment)

Adalah apapun diluar batasan sistem yang mempengaruhi operasi sistem, lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

5. Penghubung (Interface) Sistem

Merupakan media penghubung antara suatu subsistem dengan subsistem lainnya.

6. Masukkan Sistem (Input)

Adalah energi yang dimasukkan kedalam sistem, masukan dapat berupa masukkan perawatan (Maintenance Input), dan masukkan sinyal (Signal Input), maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sedangkan signal input adalah energi yang di proses untuk mendapatkan keluaran.

7. Keluaran Sistem (Output)

Adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat merupakan masukan untuk subsistem yang lain kepada supra sistem.

8. Sasaran Sistem

Suatu sistem pasti memiliki sasaran atau tujuan (Goal). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.4.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya, Menurut Al-Bahrain Bin Ladjamudin (2005:6)

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide - ide yang tidak tampak secara fisik. Sistem fisik adalah sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu dan sistem tak tentu. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat 14 diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung probabilitas.
4. Sistem diklasifikasikan sebagai sistem tertutup dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang hubungan dan terpengaruh dengan lingkungan luarnya.

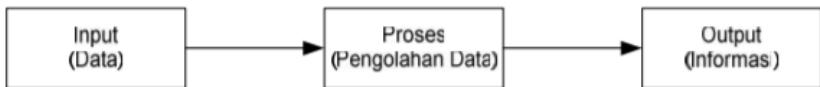
2.4.3 Analisis Sistem

Tahapan analisis sistem dimulai karena adanya permintaan terhadap sistem baru. Permintaan dapat datang dari seorang manajer dan dari luar departemen sistem informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan adanya peluang baru. Namun, adakalanya inisiatif pengembangan sistem baru berasal dari bagian yang bertanggung jawab terhadap pengembangan sistem informasi, yang bermaksud mengembangkan sistem yang sudah ada atau mengatasi masalah-masalah yang belum tertangani. Analisis sistem adalah untuk menentukan hal-hal detail tentang yang akan dikerjakan oleh sistem yang diusulkan (dan bukan bagaimana caranya). Analisis sistem mencakup studi kelayakan dan analisis kebutuhan, Abdul Kadir (2003:400).

Komponen Sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama dalam membentuk satu kesatuan komponen. Komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan [4].

2.4.4 Siklus Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, maka perlu dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Siklus informasi atau siklus pengolahan data adalah sebagai berikut :



Gambar 2.41 Siklus Informasi

2.5 Sistem yang sedang berjalan

Untuk gambaran sistem yang berjalan dalam aplikasi pengolahan data administrasi barang adalah proses dimana pengambilan barang di kantor pos Politeknik Pos Indonesia yang dibuat dengan Oracle APEX.

2.6 Dokumen yang digunakan

untuk menjelaskan dokumen yang digunakan dalam proses penyampaian informasi pada aplikasi. Berikut merupakan analisa dokumen yang menjelaskan hal - hal yang akan dilakukan dalam tabel seperti berikut.

Dibuat oleh	Admin Aplikasi
Dibuat untuk	User
Isi	List yang berisi informasi yang diperbarui atau ditambahkan.
Frekuensi	Dibuat berdasarkan informasi terbaru sesuai dengan rekam data barang.
Tujuan	Menyampaikan informasi data barang terbaru sesuai dengan rekam data barang.

Gambar 2.42 Tabel Dokumen Yang Akan Digunakan

2.7 Sistem yang Dibangun

Pada sistem ini dimaksudkan ke dalam use case mengenai sistem yang akan berjalan pada Aplikasi Pengolahan Data Administrasi Barang.

2.7.0.1 Sistem yang Akan Dibangun pada penyampaian Informasi Terbaru

Dalam aplikasi ini customer/ user mencari informasi yang akan dibutuhkan dengan cara menunjukan identitas, input kode resi barang dan untuk mengetahui barang paket telah diambil atau belum diambil harus melalui aplikasi ini menggunakan database yang telah tersedia. Sehingga kami membuat sistem pengambilan barang yang berada di kantor satpam Politeknik Pos Indonesia yang awal mulanya masih menggunakan cara manual sekarang sudah mengalami perkembangan dengan memanfaatkan kecanggihan teknologi dengan menggunakan Apex Oracle Online.

2.7.0.2 Kebutuhan Fungsional Kebutuhan fungsional adalah kebutuhan yang berhubungan dengan sistem yang berjalan. Dimana akan menjabarkan tentang fungsi-fungsi yang akan mendukung jalannya sistem. Adapun kebutuhan fungsional yang akan dibuat ada 2 yaitu berikut

1. Fungsi Search
2. Fungsi Edit
3. Fungsi Create

2.7.0.3 Kebutuhan Non-Fungsional Kebutuhan perangkat lunak dilakukan untuk mengetahui kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras (hardware) dan analisis perangkat lunak (software).

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	500 GB	Media untuk menyimpan data aplikasi yang dibuat
2	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3	<i>Processor</i>	<i>Intel® Pentium(R) CPU N4200 @1,10 Ghz</i>	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur Jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

Gambar 2.43 *Diskripsi Perangkat Keras User*

No	Tools	Fungsi	Keterangan
1	<i>Windows 10</i>	Sistem Operasi	-
3	<i>Oracle Apex</i>	desain antar muka	-
4	<i>Java</i>	<i>Bahasa Program</i>	-
5	<i>Oracle DB Express</i>	software pendukung	-
6	<i>Google Chrome</i>	<i>browser</i>	-

Gambar 2.44 *Diskripsi Perangkat Lunak User*

2.7.0.4 Kebutuhan Perangkat Keras dan Perangkat Lunak

BAB 3

LATEX

3.1 Latex

Latex merupakan bahasa markup atau sistem penyiapan dokumen untuk peranti lunak TeX. Tex merupakan program komputer yang digunakan untuk membuat typesetting suatu dokumen, atau membuat formula matematika. LaTeX memungkinkan penulis/penggunanya untuk melakukan typesetting dan mencetak hasil kerjanya dalam bentuk tipografi yang terbaik. Oleh karenanya LaTeX paling banyak digunakan oleh para matematikawan, ilmuwan, insinyur, akademisi, dan profesional lainnya. Pada awalnya LaTeX ditulis pada awal 1980-an oleh Leslie Lamport di SRI International. Versi paling mutakhir adalah LateX2e.

Dalam penggunaan Latex itu sendiri memiliki beberapa keuntungan ataupun kelebihan antaralain sebagai berikut :

Kelebihan menggunakan latex :

1. Memiliki format dokumen yang terstruktur sehingga membuat dokumen terlihat sangat profesional dan sempurna.
2. Segala jenis formula matematis dapat dituliskan dengan mudah.

3. List gambar, List tabel, Daftar Pustaka, footnote bahkan daftar isi dapat secara otomatis dibuat oleh program.
4. LaTeX melatih dan memaksa pengguna untuk membuat dokumen dengan struktur yang baik dan benar, sehingga tidak terjadi kerancuan dalam penulisan.

Kekurangan menggunakan latex :

1. Sangat sulit untuk menuliskan dokumen yang tidak terstruktur.
2. Memerlukan keredasan manusia

3.1.1 Tahapan Membuat Dokumen pada Latex

LATEX adalah sistem operasi untuk membuat dokumen. Ini didasarkan pada gagasan 'WYSIWYM' (What You See Is What You Mean), artinya Anda hanya perlu fokus pada isi dokumen Anda dan komputer akan mengurus pemformatannya. Dengan LATEX, sangat mudah untuk membuat materi yang terlihat profesional.

Berikut bagian - bagian pada latex:

1. Perkenalan
2. Pembukaan dokumen
3. Menampilkan judul dokumen Anda
4. Pemformatan dasar: abstrak, paragraf dan baris baru
5. Komentar
6. Panduan referensi

Berikut ini adalah tahapan sederhana dalam pembuatan teks di lateks.

```
\documentclass{article}
```

```
\begin{document}
```

First document. This is a simple example, with no extra parameters or packages included.

```
\end{document}
```



Gambar 3.1 Membuat dokumen pada lateks

File input hanyalah file teks biasa, dengan ekstensi .tex. Ini akan berisi kode yang ditafsirkan komputer untuk menghasilkan file PDF. Baris pertama kode menyatakan jenis dokumen, dalam hal ini adalah artikel. Kemudian, di antara tag begin document end document harus menulis teks dokumen yang akan dibuat.

3.1.2 Pembukaan Dalam Pembuatan Dokumen

Pada contoh sebelumnya, teks dimasukkan setelah perintah begin document. Bagian dari file .tex Anda sebelum titik ini disebut pembukaan. Dalam pembukaan, Anda menentukan jenis dokumen yang Anda tulis dan bahasanya, memuat paket tambahan yang Anda perlukan, dan menetapkan beberapa parameter. Misalnya, pembukaan dokumen normal akan terlihat seperti ini:

```
\documentclass[12pt, letterpaper]{article}
\usepackage[utf8]{inputenc}

\title{First document}
\author{Database \thanks{funded by the Overleaf team}}
\date{February 2020}
```

Gambar 3.2 Pembukaan dokumen pada lateks

Di bawah ini penjelasan terperinci untuk setiap baris:

documentclass [12pt, letterpaper] article Seperti yang dikatakan sebelumnya, ini mendefinisikan jenis dokumen. Beberapa parameter tambahan di dalam tanda kurung dan dipisahkan koma dapat diteruskan ke perintah. Dalam contoh tersebut, parameter tambahan mengatur ukuran font (12pt) dan ukuran kertas (kertas surat). Tentu saja ukuran font lain (9pt, 11pt, 12pt) dapat digunakan, ukuran standarnya adalah 10pt. Adapun ukuran kertas, nilai-nilai lain yang mungkin termasuk A4 dan kertas legal. Perhatikan bahwa Overleaf menggunakan distribusi LaTeX Eropa, yang secara default menghasilkan dokumen dalam ukuran A4. Lihat [ukuran halaman dan margin]] artikel kami untuk informasi tentang menggunakan paket Geometri untuk mengatur parameter pemformatan lebih lanjut. usepackage [utf8] inputenc Ini adalah penyandian untuk dokumen, untuk memungkinkan karakter di luar ASCII (mis. , , ...) untuk digunakan dalam teks. Itu dapat dihilangkan atau diubah ke penyandian lain tetapi utf-8 direkomendasikan. Kecuali jika Anda secara khusus memerlukan penyandian lain, atau jika Anda tidak yakin tentang hal itu, tambahkan baris ini ke pembukaan. Tiga baris berikutnya adalah deskriptif diri. Bagaimanapun, Anda dapat melihat deskripsi tentang apa yang sebenarnya mereka lakukan di bagian selanjutnya. Parameter penting lain yang dapat diteruskan ke perintah documentclass adalah twocolumn jika Anda ingin teks Anda dalam format dua kolom dan dua sisi untuk pencetakan lembaran kertas dua sisi.

3.1.3 Menampilkan judul dokumen

Untuk menampilkan judul dokumen anda, anda harus mendeklarasikan komponennya di pembukaan dan kemudian menggunakan beberapa kode tambahan:

```
\documentclass[12pt, letterpaper, twoside]{article}
\usepackage[utf8]{inputenc}

\title{First document}
\author{Hubert Farnsworth \thanks{funded by the Overleaf team}}
\date{February 2014}

\begin{document}

\begin{titlepage}
\maketitle
\end{titlepage}

In this document some extra packages and parameters
were added. There is an encoding package
and pagesize and fontsize parameters.

\end{document}
```

Gambar 3.3 Menampilkan dokumen pada lateks



Gambar 3.4 Tampilan dokumen pada latex

Ada satu blok dengan tiga baris di pembukaan yang mendefinisikan informasi untuk dimasukkan pada halaman judul.

```
\title{Dokumen pertama}
```

Ini judulnya.

```
\author{Hubert Farnsworth}
```

Di sini Anda meletakkan nama penulis dan, sebagai parameter opsional, Anda dapat menambahkan perintah selanjutnya:

```
\thanks{didanai oleh tim Proyek 2}
```

Ini dapat ditambahkan setelah nama penulis, di dalam kurung perintah judul. Ini akan menambahkan superskrip dan catatan kaki dengan teks di dalam kurung kurawal. Berguna jika Anda perlu mengucapkan terima kasih kepada lembaga di artikel Anda.

```
\date{Februari 2020}
```

Anda dapat memasukkan tanggal secara manual atau menggunakan perintah *September 30, 2020* sehingga tanggal akan diperbarui secara otomatis pada saat anda menyusun dokumen Anda. Setelah anda memilikinya di pembukaan sekarang di tubuh dokumen Anda, Anda dapat menggunakan perintah berikutnya untuk informasi yang akan dicetak.

```
\begin{titlepage}  
\end{titlepage}
```

Ini menyatakan lingkungan, blok kode dengan perilaku tertentu tergantung pada jenisnya. Dalam hal ini apa pun yang Anda sertakan dalam lingkungan titlepage ini akan muncul di halaman pertama dokumen Anda.

```
\maketitle
```

Perintah ini akan mencetak judul, penulis dan tanggal dalam format yang ditunjukkan pada contoh. Jika tidak terlampir di lingkungan titlepage, itu akan ditampilkan di awal dokumen, di atas baris pertama.

3.1.4 Format Dasar: Abstrak, Paragraf, dan Baris baru

Segala sesuatu yang tercakup dalam perintah `begin document` `end document` akan disajikan dalam dokumen akhir.

```
\documentclass[12pt, letterpaper, twoside]{article}
\usepackage[utf8]{inputenc}

\begin{document}

\begin{abstract}
This is a simple paragraph at the beginning of the document. A brief introduction to the main subject.
\end{abstract}

In this document some extra packages and parameters
were added. There is an encoding package,
and pagesize and fontsize parameters.

This line will start a second paragraph. And I can
break\\ the lines \\ and continue on a new line.

\end{document}
```

Gambar 3.5 *Absbtrak paragraf baru*

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

In this document some extra packages and parameters were added. There is a encoding package an a pagesize and fontsize parameters.

This line will start a second Paragraph. And I can break the lines
and continue in a new line.

Gambar 3.6 *Contoh dari abstrak*

Dalam dokumen ilmiah, ini adalah praktik umum untuk memasukkan tinjauan singkat tentang subjek utama makalah ini. Dalam LATEX, lingkungan abstrak adalah untuk tujuan ini. Lingkungan abstrak akan menempatkan teks dalam format khusus di bagian atas dokumen Anda.

Saat menulis konten dokumen Anda, jika Anda harus memulai paragraf baru, Anda harus menekan tombol "Enter" dua kali (untuk menyisipkan baris kosong ganda). Perhatikan bahwa paragraf memiliki spasi putih sebelum baris pertama.

Untuk memulai baris baru tanpa benar-benar memulai paragraf baru, masukkan break line point, ini dapat dilakukan oleh (double backslash seperti pada contoh) atau perintah newline

3.1.5 Kolom Komentar

Terkadang perlu menambahkan komentar ke kode LATEX Anda agar mudah dibaca. Ini mudah, letakkan % sebelum komentar dan LATEX akan mengabaikan teks itu.

```
\documentclass{article}
\usepackage[utf8]{inputenc} %codification of the document

\usepackage{comment}

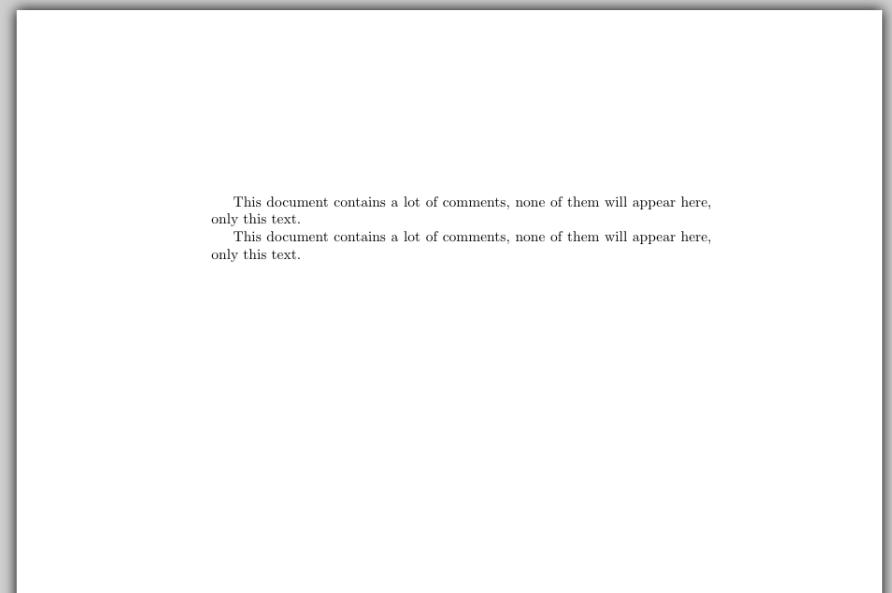
%Here begins the body of the document
\begin{document}
This document contains a lot of comments, none of them
will appear here, only this text.

This document contains a lot of comments, none of them
will appear here, only this text.

\begin{comment}
This text won't show up in the compiled pdf
this is just a multi-line comment. Useful
to, for instance, comment out slow-rendering parts
while working on a draft.
\end{comment}

\end{document}
```

Gambar 3.7 Pembuatan pada kolom komentar 1



```
This document contains a lot of comments, none of them will appear here,  
only this text.
```

```
This document contains a lot of comments, none of them will appear here,  
only this text.
```

Gambar 3.8 *Hasil pemuatan kolom komentar*

Di bagian terakhir contoh, Anda dapat melihat lingkungan komentar. Ini membantu dalam komentar multi-baris alih-alih menempatkan tanda persen di awal setiap baris. Agar ini berfungsi, Anda harus menambahkan baris berikutnya ke mukadimah Anda:

usepackage comment Simbol persen adalah karakter yang dipesan, jika Anda benar-benar membutuhkan simbol ini untuk dicetak dalam dokumen Anda, gunakan persen. Lihat panduan referensi untuk daftar lengkap karakter yang dipesan.

3.1.6 Panduan referensi

Dibawah ini merupakan Jenis dokumen tersedia dalam perintah atau documentclass.

Jenis dokumen	Keterangan
article	Untuk dokumen pendek dan artikel jurnal. Merupakan yang paling umum digunakan.
report	Untuk dokumen dan disertasi yang lebih panjang.
book	book Berguna untuk menulis buku
letter	Untuk surat
slides	Untuk slide, jarang digunakan
beamer	Slides dalam format kelas Beamer. Lihat dokumentasi beamer untuk deskripsi yang lebih baik

Gambar 3.9 Panduan referensi dalam pembuatan lateks

3.1.7 Karakter

Karakter simbol berikut dicadangkan oleh LATEX karena mereka memperkenalkan perintah dan memiliki makna khusus.Simbol-simbol ini dan dapat dicetak dengan perintah khusus (dalam beberapa kasus - di dalam lingkungan matematika) antara lain sebagai berikut.

Jenis dokumen	Keterangan
article	Untuk dokumen pendek dan artikel jurnal. Merupakan yang paling umum digunakan.
report	Untuk dokumen dan disertasi yang lebih panjang.
book	book Berguna untuk menulis buku
letter	Untuk surat
slides	Untuk slide, jarang digunakan
beamer	Slides dalam format kelas Beamer. Lihat dokumentasi beamer untuk deskripsi yang lebih baik

Gambar 3.10 Panduan referensi dalam karakter dalam lateks

Karakter	Fungsi	Penggunaan
#	Parameter makro	\#
\$	Math mode	\$
%	Komentar	\%
[^]	Superscript (dalam mode matematika)	\^{} or \textasciicircum
&	Pisahkan entri kolom dalam tabel	\&
-	Subskrip (dalam mode matematika)	_
{}	Memproses blok	\{\}
\sim	Ruang tidak bisa dipecahkan, gunakan setiap kali Anda ingin meninggalkan ruang yang tidak bisa dipecahkan	\textasciitilde or \sim{}
\	Mulai perintah, yang diperlukan hingga karakter non-alfanumerik pertama	\textbackslash or \

Gambar 3.11 Panduan referensi dalam karakter dalam lateks

3.1.8 Memilih Kompiler LaTeX

Gaya, konten, dan tata letak dalam dokumen LATEX ditentukan dengan cara tag atau perintah dalam file .tex biasa, file ini dapat digunakan untuk menghasilkan beberapa jenis versi dokumen yang "dapat dibaca manusia". Cara termudah untuk menghasilkan hasil akhir ini adalah menggunakan Overleaf. Overleaf memiliki distribusi LATEX yang siap digunakan dan dokumen final dapat dihasilkan dengan mengklik tombol "Kompilasi ulang"; jika ini adalah kasus Anda, Anda dapat melewati artikel ini. Kalau tidak, jika Anda perlu mempelajari cara mengkompilasi dokumen di komputer Anda, artikel ini menjelaskan cara menghasilkan output PS, DVI dan PDF dari file LATEX. Adapun isinya sebagai berikut ini :

1. Perkenalan
2. Distribusi TeX
3. Editor LaTeX
4. Format keluaran
5. Kompiler lain
6. Menyusun dokumen dengan referensi silang
7. Panduan referensi

Dokumen LATEX adalah dokumen biasa dengan ekstensi .tex (lihat artikel Membuat dokumen di LaTeX untuk contoh), file teks biasa ini memiliki beberapa perintah markup yang dimaksudkan untuk memformat dokumen tetapi, bagaimana Anda benar-benar menghasilkan hasil akhir? Itu tergantung pada jenis dokumen yang ingin Anda hasilkan.

Misalkan Anda menyimpan dokumen Anda dan menamai file "mydocument.tex". Untuk membuat file .PDF jalankan perintah ini di terminal sistem.

```
pdflatex mydocument.tex
```

Dan sebuah file bernama "mydocument.pdf" akan muncul.

3.1.9 Distribusi TeX

Serangkaian program yang memungkinkan untuk mengkompilasi dokumen TEX dan LATEX disebut pengaturan huruf TEX atau distribusi TEX. Ada banyak distribusi TEX yang tersedia untuk sistem operasi yang berbeda:

1. MiKTeX untuk Windows
2. TeX Live untuk Linux dan sistem mirip UNIX lainnya
3. Redistribusi MacTeX dari TeX Live untuk macOS
4. Redistribusi MacTeX dari TeX Live untuk macOS
5. teTeX untuk Linux dan sistem mirip UNIX lainnya; itu tidak lagi dipelihara secara aktif sekarang
6. proTeXt didasarkan pada MiKTeX

3.1.10 Editor Latex

Ada banyak editor teks canggih yang khusus didedikasikan untuk LATEX untuk sistem operasi paling populer, beberapa di antaranya dapat diunduh secara gratis sementara yang lain adalah perangkat lunak berpemilik:

1. Sumber terbuka: AUCTEX, GNU TeXmacs, Gummi, Kile, LaTeXila, MeWa, TeXShop, TeXnicCenter, Texmaker, TeXstudio, TeXworks
2. Freeware: LEd, WinShell
3. Proprietary / Shareware: Inlage, Scientific WorkPlace, WinEdt

Ada juga editor untuk keperluan umum yang menambahkan opsi ramah LATEX melalui plugin. Misalnya editor emacs dan vim yang terkenal memiliki keduanya Templat: ekstensi LaTeX, seperti halnya Atom, Kode VS dan Teks Sublim.

Perlu disebutkan bahwa editor di Overleaf dapat dikustomisasi untuk meniru perilaku emacs atau vim. Ini juga dapat disesuaikan, sehingga Anda dapat memilih

penyorotan sintaksis yang sama dari beberapa editor yang disebutkan sebelumnya. Anda dapat menemukan pengaturan ini dengan mengklik ikon menu Overleaf, di atas panel daftar file dalam proyek Overleaf Anda.

3.1.11 Format Keluaran

Ada tiga format output yang tersedia di semua distribusi TEX. Untuk menghasilkan keluaran tertentu, dokumen harus dikompilasi dengan menjalankan perintah di terminal (Linux dan macOS) atau prompt perintah (windows).

Misalnya, jika Anda ingin mengompilasi file bernama "mydocument.tex" Anda dapat menggunakan salah satu dari perintah berikut:

lateks mydocument.tex Ini akan membuat "mydocument.dvi", dokumen DVI pdflatex mydocument.tex Ini akan menghasilkan "mydocument.pdf", dokumen PDF Ada beberapa perintah tambahan yang memungkinkan konversi antar format, gambar PS juga didukung. Lihat panduan referensi untuk deskripsi skematik dari perintah dan format output.

Deskripsi setiap format output disediakan di bawah ini:

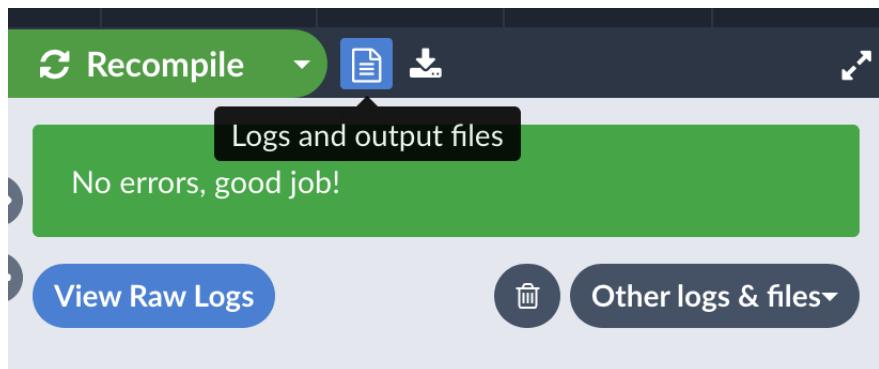
1. (DVI) Format file independen perangkat terdiri dari data biner yang menggambarkan tata letak visual dokumen dengan cara yang tidak bergantung pada format gambar tertentu, perangkat keras display, atau printer.
2. (PS) Format file PostScript menggambarkan teks dan grafik pada halaman dan didasarkan pada grafik vektor. PostScript, sampai sekarang, merupakan standar dalam area penerbitan desktop.
3. (PDF) Portable Document Format adalah format file, berdasarkan PostScript, digunakan untuk mewakili dokumen dengan cara yang tidak tergantung pada perangkat lunak aplikasi, perangkat keras, dan sistem operasi. Sekarang banyak digunakan sebagai format file untuk dicetak dan untuk distribusi di Web.

Sebagian besar editor saat ini menyertakan ikon akses cepat untuk dikompilasi ke format output yang berbeda sehingga Anda tidak harus benar-benar menjalankan perintah dalam prompt sistem.

Di Overleaf, hasil akhir yang dikompilasi selalu berupa dokumen PDF. Jika Anda diminta untuk menghasilkan file DVI dari proyek Overleaf Anda, inilah yang dapat Anda lakukan seperti berikut ini:

1. Pastikan Anda hanya menggunakan gambar .ep dan .ps di proyek Anda.
2. Klik pada ikon menu Overleaf di atas panel daftar file, dan atur pengaturan Compiler ke 'LaTeX'.
3. Kompilasi ulang proyek Anda.
4. Klik tombol "Log dan output file" di sebelah tombol Recompile.
5. Gulir ke kanan ke bawah, dan klik "Log lain dan file output".

6. Anda kemudian dapat mengunduh file .dvi yang dihasilkan.



Gambar 3.12 Format keluaran dalam lateks

3.1.12 Kompiler lain

Pengaturan kompiler lain yang mungkin adalah pdfLaTeX (default), XeLaTeX dan LuaLaTeX. Anda biasanya dapat menggunakan pdfLaTeX, tetapi memilih kompiler tergantung pada kebutuhan masing-masing proyek.

1. LaTeX hanya mendukung format gambar .eps dan .ps untuk digunakan dengan `Includedgraphics`. Jika semua gambar dalam proyek Anda adalah file .eps, maka pengaturan kompiler ini direkomendasikan.
2. pdfLaTeX mendukung format gambar .png, .jpg, .pdf. Ini akan mengonversi gambar .eps menjadi .pdf sambil jalan saat kompilasi, yang dapat memperpanjang waktu kompilasi yang diperlukan. (pdfLaTeX mungkin tidak dapat menangani pstricks dengan baik di Overleaf.)
3. XeLaTeX dan LuaLaTeX keduanya mendukung UTF-8 dengan kuat di luar kotak, serta Truetype dan OpenType. Karena itu mereka direkomendasikan jika Anda perlu mengeset skrip non-Latin di Overleaf, bersama dengan pacakge `polyglossia`. Mereka juga mendukung semua format gambar .png, .jpg, .pdf dan .eps.
4. XeLaTeX mendukung pstricks; tapi LuaLaTeX tidak.
5. Anda dapat memperluas kemampuan LuaLaTeX dengan menyematkan kode Lua langsung ke dokumen.

3.1.13 Menyusun dokumen dengan referensi silang

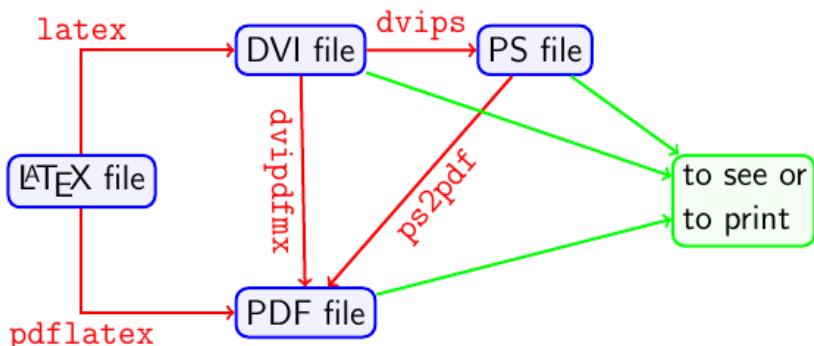
Dalam beberapa kasus, ketika dokumen Anda menyertakan referensi silang, Anda harus mengkompilasi sumbernya dua kali. Penting untuk memasukkan angka yang benar dalam daftar isi, daftar gambar, nomor referensi untuk teorema dan sebagainya.

Selama kompilasi pertama kompiler LATEX menulis file .aux untuk informasi tentang penomoran yang berbeda dan selama yang kedua kompiler membaca informasi ini untuk menghasilkan daftar isi, bibliografi, dll.

Proses ini dapat diotomatisasi oleh perintah latexmk. Misalnya, untuk membuat pdf dari file "mydocument.tex", jalankan

```
latexmk -pdf mydocument.tex
```

hanya sekali, bahkan jika dokumen tersebut telah merujuk gambar dan daftar pustaka. Anda dapat mengubah -pdf ke -dvi untuk menetapkan jenis file yang berbeda.



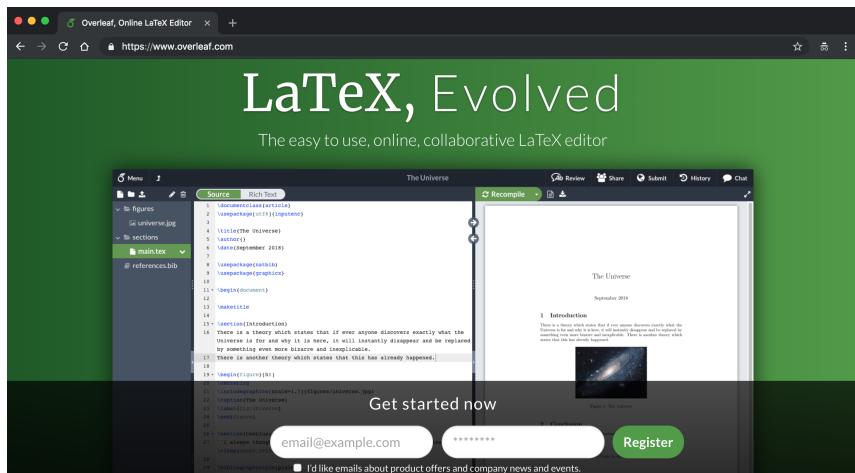
Gambar 3.13 Penyusunan dokumen

3.1.14 Membuat dokumen di Overleaf

Overleaf adalah alat edisi LaTeX online yang hebat yang memungkinkan Anda membuat dokumen LaTeX secara langsung di peramban web Anda. Artikel ini menjelaskan cara membuat proyek baru di Overleaf, baik mulai dari awal, mengunggah file Anda sendiri atau menggunakan salah satu dari banyak templat yang tersedia.

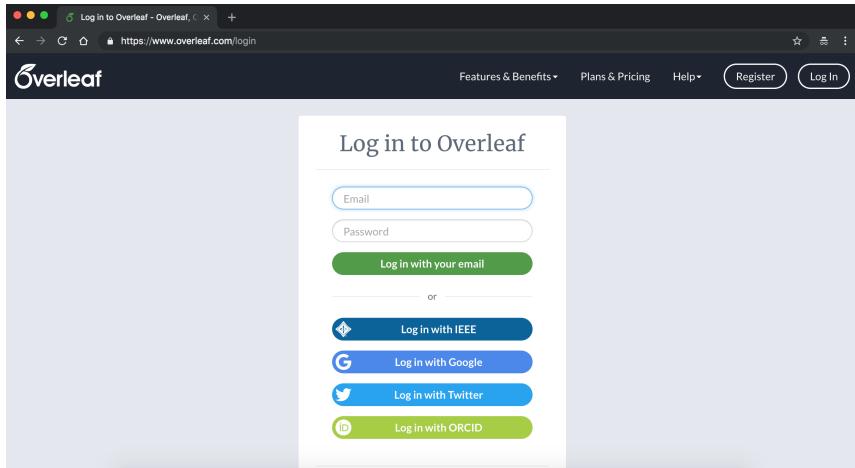
Untuk mulai menggunakan Overleaf, buka www.overleaf.com. Jika Anda tidak memiliki akun, masukkan alamat email Anda dan atur kata sandi di kotak yang sesuai di bawah. Mulailah sekarang, klik Daftarkan dan hanya itu, Anda akan diarahkan ke halaman manajemen proyek di mana Anda akan dipandu ke bagaimana buat proyek baru.

- Untuk mulai menggunakan Overleaf, buka www.overleaf.com. Jika Anda tidak memiliki akun, masukkan alamat email Anda dan atur kata sandi di kotak yang sesuai di bawah. Mulailah sekarang, klik Daftarkan dan hanya itu, Anda akan diarahkan ke halaman manajemen proyek di mana Anda akan dipandu ke bagaimana membuat proyek baru.



Gambar 3.14 User interface dalam lateks

- Jika Anda sudah memiliki akun, klik Login di sudut kanan atas, lalu ketikkan email dan kata sandi Anda dan klik tombol Login.



Gambar 3.15 User interface dalam login lateks

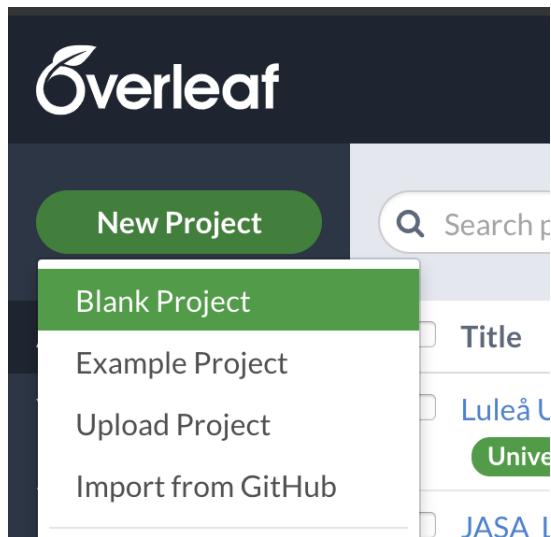
3. Setelah Anda masuk, Anda akan melihat halaman Manajemen Proyek Overleaf

The screenshot shows the Overleaf web interface. On the left, there's a sidebar with options like 'New Project', 'All Projects', 'Your Projects', 'Shared with you', 'Archived Projects', 'Trashed Projects', 'TAGS/FOLDERS', and '+ New Folder'. Below that is a section asking if the user is affiliated with an institution. The main area is titled 'Search projects...' and lists several documents with their titles, owners, and last modified times:

Title	Owner	Last Modified
ISBN HIYA HIYA	You	a few seconds ago by You
First document (1)	You	3 hours ago by You
First document	You	3 hours ago by You
yao0[You	2 days ago by You
ISBN	You	2 days ago by You
metlid	You	a month ago by You
proyek2	You	a month ago by You
untitled (1)	You	a month ago by You

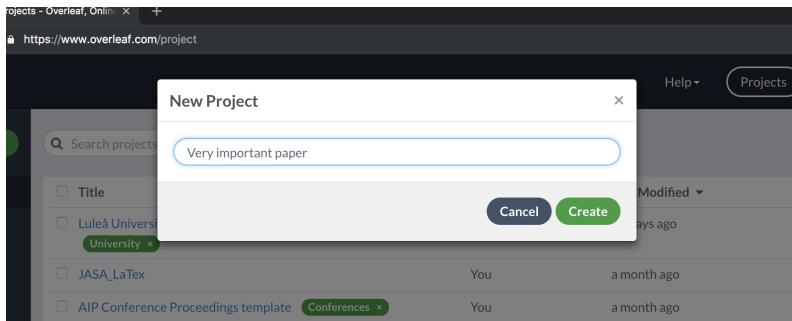
Gambar 3.16 File yang tersimpan pada lateks

4. Memulai Project, Untuk memulai proyek baru dari awal, di halaman utama klik tombol Proyek Baru, Anda akan melihat menu drop-down berikutnya



Gambar 3.17 User interface dalam membuat proyek baru

5. Lalu klik Projek Kosong. Sebuah kotak akan terbuka di mana Anda harus memasukkan nama proyek baru Anda, lalu klik Buat.



Gambar 3.18 Pembuatan objek baru

6. Setelah itu Anda akan diarahkan ke editor.

```
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\usepackage{comment}

\usepackage[
backend=biber,
style=alphabetic,
sorting=nyt
]{biblatex}
\addbibresource{sample.bib}

\title{Bibliography management: \texttt{biblatex} package}
\author{Overleaf}
\date{ }

\begin{document}

\maketitle

Using \texttt{biblatex} you can display bibliography divided into sections, depending of citation type.
Let's cite! The Einstein's journal paper \cite{einstein} and the Dirac's book \cite{dirac} are physics related items.
Next, \textit{The \LaTeX\ Companion} book \cite{latexcompanion}, the Donald Knuth's website \cite{knuthwebsite}, \textit{The Comprehensive Tex Archive Network} (CTAN) \cite{ctan} are \LaTeX\ related items; but the others Donald Knuth's items \cite{knuth-fa,knuth-acp} are dedicated to programming.

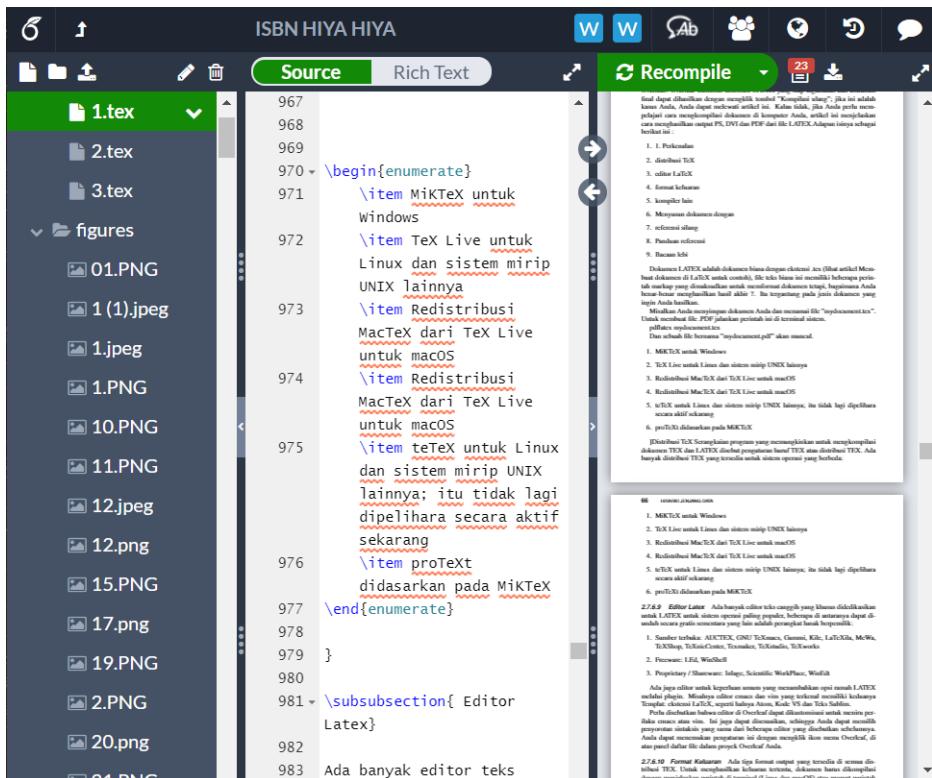
\medskip

\printbibliography[title={Whole bibliography}]
```

Gambar 3.19 Editor dalam lateks

Di sana, dokumen baru dibuat dengan beberapa informasi dasar yang sudah diisi. Anda dapat mulai mengedit file .tex Anda sekarang, untuk melihat perubahannya klik Kompilasi ulang. Untuk informasi lebih lanjut tentang cara membuat dokumen LATEX dasar lihat Membuat dokumen di LaTeX.

7. Pilihan lain untuk memulai proyek baru dari awal adalah dengan menggunakan Proyek Contoh. Di halaman Manajemen Proyek klik tombol Proyek baru (gambar pertama di bagian ini) dan di menu drop-down, klik Contoh Proyek, Anda akan melihat kotak teks di mana Anda harus memasukkan nama proyek baru Anda (gambar kedua di bagian ini), sekarang klik Buat dan Anda akan diarahkan ke editor tempat dokumen sampel akan terbuka



Gambar 3.20 Contoh dokumen pada lateks

3.2 Panduan Cepat

Panduan cepat untuk membuat proyek kosong baru: Di halaman Manajemen Proyek, klik Proyek lalu Proyek Kosong, masukkan nama untuk proyek Anda, lalu klik Buat.

Panduan cepat untuk membuat proyek baru dari sebuah contoh: Di halaman Manajemen Proyek, klik Proyek lalu Contoh Proyek, masukkan nama untuk proyek Anda, lalu klik Buat.

3.2.1 Manajemen bibliografi di LaTeX

Ketika datang ke paket manajemen bibliografi, ada tiga opsi utama dalam LATEX: bibtex, natbib (paket untuk digunakan dengan bibtex) dan biblatex. Biblatex adalah program modern untuk memproses informasi bibliografi, menyediakan antarmuka yang lebih mudah dan lebih fleksibel dan lokalisasi bahasa yang lebih baik daripada dua opsi lainnya. Artikel ini menjelaskan cara menggunakan biblatex untuk men-gelola dan memformat bibliografi dalam dokumen LATEX.

Contoh paket biblatex yang berfungsi minimal ditunjukkan di bawah ini:

```
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\usepackage{biblatex}
\addbibresource{sample.bib}

\begin{document}
Let's cite! The Einstein's journal paper \cite{einstein} and the Dirac's book \cite{dirac} are physics related items.

\printbibliography

\end{document}
```

Gambar 3.21 Contoh penggunaan Biblatex

Let's cite! The Einstein's journal paper [2] and the Dirac's book [1] are physics related items.

References

- [1] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.
- [2] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.

Gambar 3.22 Bibilatex

Ada empat perintah terkait bibliografi dalam contoh ini:

```
\usepackage{biblatex}
```

Impor paket biblatex.

```
\addbibresource{sample.bib}
```

Impor sampel file data bibtex.bib, file ini adalah yang memuat informasi tentang setiap buku yang direferensikan, artikel, dll. Lihat bagian file bibliografi untuk informasi lebih lanjut.

```
\cite{einstein}
```

Perintah ini memasukkan referensi di dalam dokumen, [1] dalam hal ini, yang sesuai dengan elemen dalam bibliografi, "einstein" adalah kata kunci yang sesuai dengan entri dalam sample.bib.

```
\printbibliography
```

Mencetak daftar referensi yang dikutip, judul defaultnya adalah "Referensi" untuk kelas dokumen artikel dan "Bibliografi" untuk buku dan laporan.

3.2.2 Penggunaan dasar

Beberapa parameter dapat dikirimkan ke perintah impor paket, seperti dalam contoh berikut:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\usepackage{comment}

\usepackage[
backend=biber,
style=alphabetic,
sorting=ynt
]{biblatex}
\addbibresource{sample.bib}

\titl{Bibliography management: \texttt{biblatex} package}
\author{Overleaf}
\date{ }

\begin{document}

\maketitle

Using \texttt{biblatex} you can display bibliography divided into sections, depending of citation type.
Let's cite! Einstein's journal paper \cite{einstein} and the Dirac's book \cite{dirac} are physics related items.
Next, \textit{The \LaTeX\ Companion} book \cite{latexcompanion}, the Donald Knuth's website \cite{knuthwebsite}, \textit{The Comprehensive Tex Archive Network} (CTAN) \cite{ctan} are \LaTeX\ related items; but the others Donald Knuth's items \cite{knuth-fa,knuth-acp} are dedicated to programming.

\medskip

\printbibliography

\end{document}
```

Gambar 3.23 Parameter dasar dalam lateks

Using **biblatex** you can display bibliography divided into sections, depending of citation type. Let's cite! The Einstein's journal paper [Ein05] and the Dirac's book [Dir81] are physics related items. Next, *The L^AT_EX Companion* book [GMS93], the Donald Knuth's website [Knu], *The Comprehensive Tex Archive Network* (CTAN) [Gre93] are L^AT_EX related items; but the others Donald Knuth's items [Knu73; Knu68] are dedicated to programming.

References

- [Ein05] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.
- [Knu68] Donald E. Knuth. *The Art of Computer Programming*. Four volumes. Seven volumes planned. Addison-Wesley, 1968.
- [Knu73] Donald E. Knuth. "Fundamental Algorithms". In: Addison-Wesley, 1973. Chap. 1.2.
- [Dir81] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.
- [GMS93] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.
- [Gre93] George D. Greenwade. "The Comprehensive Tex Archive Network (CTAN)". In: *TUGBoat* 14.3 (1993), pp. 342–351.
- [Knu] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>. (accessed: 01.09.2016).

Gambar 3.24 Penggunaan *biblatex*

Beberapa opsi tambahan, di dalam tanda kurung dan dipisahkan dengan koma, ditambahkan saat mengimpor biblatex:

```
backend = biber
```

Mengatur backend untuk mengurutkan bibliografi, biber adalah yang default dan direkomendasikan karena memberikan pelokalan penuh untuk beberapa perintah dan gaya untuk biber lebih mudah untuk dimodifikasi karena mereka menggunakan makro LATEX standar. Backend yang didukung lainnya adalah bibtex, yang merupakan program yang lebih tradisional; jika diatur sebagai backend, bibtex hanya akan digunakan untuk mengurutkan daftar pustaka, sehingga tidak ada gaya bibtex yang dapat digunakan di sini.

```
style = alfabet
```

Menentukan gaya bibliografi dan gaya kutipan, dalam hal ini adalah abjad. Bergantung pada gayanya, lebih banyak perintah kutipan mungkin tersedia.

```
sorting = ynt
```

Menentukan kriteria untuk mengurutkan sumber bibliografi. Dalam hal ini mereka diurutkan berdasarkan tahun, nama dan judul.

3.2.3 The bibliography file

File bibliografi harus memiliki sintaks bibtex standar

```
@article{einstein,
    author = "Albert Einstein",
    title = "{Zur Elektrodynamik bewegter K\"orper}. ({German})
              [{On} the electrodynamics of moving bodies]",
    journal = "Annalen der Physik",
    volume = "322",
    number = "10",
    pages = "891--921",
    year = "1905",
    DOI = "http://dx.doi.org/10.1002/andp.19053221004",
    keywords = "physics"
}

@book{dirac,
    title = {The Principles of Quantum Mechanics},
    author = {Paul Adrien Maurice Dirac},
    isbn = {9780198520115},
    series = {International series of monographs on physics},
    year = {1981},
    publisher = {Clarendon Press},
    keywords = {physics}
}

@online{knuthwebsite,
    author = "Donald Knuth",
    title = "Knuth: Computers and Typesetting",
    url = "http://www-cs-faculty.stanford.edu/~uno/abcde.html",
    addendum = "(accessed: 01.09.2016)",
    keywords = "latex,knuth"
}

@inbook{knuth-fa,
    author = "Donald E. Knuth",
    title = "Fundamental Algorithms",
    publisher = "Addison-Wesley",
    year = "1973",
    chapter = "1.2",
    keywords = "knuth,programming"
}
...
}
```

Gambar 3.25 syntax pada biblatex

File ini berisi catatan dalam format khusus, misalnya, referensi bibliografi pertama ditentukan oleh:

```
@artikel{...}
```

Ini adalah baris pertama dari entri catatan, @article memberitahu BibTeX bahwa informasi yang disimpan di sini adalah tentang sebuah artikel. Informasi tentang entri ini terlampir dalam kurung kurawal. Selain jenis entri yang ditunjukkan dalam contoh (artikel, buku, online dan inbook) ada banyak lagi, lihat panduan referensi.

```
Einstein
```

Label Einstein ditugaskan untuk entri ini, adalah pengidentifikasi unik yang dapat digunakan untuk merujuk artikel ini dalam dokumen.

```
author = "Albert Einstein",
```

Ini adalah bidang pertama dalam entri bibliografi, menunjukkan bahwa penulis artikel ini adalah Albert Einstein. Beberapa bidang yang dipisahkan koma dapat ditambahkan menggunakan kunci sintaks yang sama = nilai, misalnya: judul, halaman, tahun, URL, dll. Lihat panduan referensi untuk daftar bidang yang mungkin. Informasi dalam file ini nantinya dapat dicetak dan direferensikan dalam dokumen LATEX, seperti yang ditunjukkan pada bagian sebelumnya, dengan perintah `adbibresource sample.bib`. Tidak semua informasi dalam file `.bib` akan ditampilkan, itu tergantung pada gaya bibliografi yang ditetapkan dalam dokumen.

3.2.4 Menyesuaikan daftar pustaka

Biblatex memungkinkan penyesuaian tinggi bagian bibliografi dengan sedikit usaha. Disebutkan bahwa beberapa gaya kutipan dan gaya bibliografi tersedia, dan Anda juga dapat membuat yang baru. Opsi penyesuaian lain adalah mengubah judul default bagian bibliografi.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\usepackage{comment}

\usepackage[
backend=biber,
style=alphabetic,
sorting=ynt
]{biblatex}
\addbibresource{sample.bib}

\titl{Bibliography management: \texttt{biblatex} package}
\author{Overleaf}
\date{ }

\begin{document}

\maketitle

Using \texttt{biblatex} you can display bibliography divided into sections, depending of citation type.  

Let's cite! The Einstein's journal paper \cite{einstein} and the Dirac's book \cite{dirac} are physics related items.  

Next, \textit{The \LaTeX\ Companion} book \cite{latexcompanion}, the Donald Knuth's website \cite{knuthwebsite}, \textit{The Comprehensive Tex Archive Network} (CTAN) \cite{ctan} are \LaTeX\ related items; but the others Donald Knuth's items \cite{knuth-fa,knuth-acp} are dedicated to programming.

\medskip

\printbibliography[title={Whole bibliography}]
```

Gambar 3.26 Penyesuaian biblatex pada daftar pustaka

Using `biblatex` you can display bibliography divided into sections, depending of citation type. Let's cite! The Einstein's journal paper [Ein05] and the Dirac's book [Dir81] are physics related items. Next, *The L^AT_EX Companion* book [GMS93], the Donald Knuth's website [Knu], *The Comprehensive Tex Archive Network* (CTAN) [Gre93] are L^AT_EX related items; but the others Donald Knuth's items [Knu73; Knu68] are dedicated to programming.

Whole bibliography

- [Ein05] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.
- [Knu68] Donald E. Knuth. *The Art of Computer Programming*. Four volumes. Seven volumes planned. Addison-Wesley, 1968.
- [Knu73] Donald E. Knuth. "Fundamental Algorithms". In: Addison-Wesley, 1973. Chap. 1.2.
- [Dir81] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.
- [GMS93] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.
- [Gre93] George D. Greenwade. "The Comprehensive Tex Archive Network (CTAN)". In: *TUGBoat* 14.3 (1993), pp. 342–351.
- [Knu] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.

Gambar 3.27 Penggunaan `biblatex`

Judul parameter tambahan = Bibliografi lengkap yang dilewatkan di dalam tanda kurung ke perintah printbibliografi adalah salah satu yang mengubah judul.

Daftar pustaka juga dapat dibagi menjadi beberapa bagian berdasarkan filter yang berbeda, misalnya: hanya mencetak referensi dari penulis yang sama, jurnal yang sama atau judul yang serupa. Di bawah ini sebuah contoh.

```
\printbibliography[type=article,title={Articles only}]\printbibliography[type=book,title={Books only}]  
\printbibliography[keyword={physics},title={Physics-related only}]\printbibliography[keyword={latex},title={\LaTeX-related only}]
```

Gambar 3.28 parameter pada lateks

Di sini, daftar pustaka dibagi dalam 4 bagian. Sintaks dari perintah yang digunakan di sini dijelaskan di bawah ini:

printbibliography [type = article, title = Artikel saja] Hanya mencetak entri yang tipenya "artikel", dan menetapkan judul "Artikel saja" untuk bagian ini. Sintaks yang sama berfungsi untuk semua tipe entri lainnya. printbibliography [kata kunci = fisika, judul = Hanya yang berhubungan dengan fisika] Memfilter entri bibliografi yang menyertakan kata "fisika" di salah satu bidang. Setel judul "Hanya yang terkait dengan Fisika" untuk bagian tersebut.

3.2.5 Menambahkan daftar pustaka dalam daftar isi

Untuk daftar pustaka yang akan dicetak dalam daftar isi, opsi tambahan harus diberikan ke printbibliography

```
\printbibliography[heading=bibintoc,title={Whole bibliography}]\printbibliography[heading=subbibintoc,type=article,title={Articles only}]
```

Gambar 3.29 Penambahan daftar pustaka dalam daftar isi di biblatex

Contents

1 First section	1
Whole bibliography	1
Articles only	3

Gambar 3.30 *Tampilan setelah penggunaan biblatex di lateks*

Bagian dan subbagian ditambahkan ke daftar isi:

1. Dalam kasus pertama, menambahkan heading = bibintoc menambahkan judul ke daftar isi sebagai bab yang tidak dinomori jika mungkin atau sebagai bagian yang tidak dinomori.
2. Kasus kedua adalah heading = subbibintoc yang menambahkan judul sebagai entri level kedua dalam daftar isi, dalam contoh ini sebagai subbagian yang bersarang di "Seluruh bibliografi".

BAB 4

MODEL BASIS DATA

Model basis data merupakan sekumpulan dari berbagai konsep basis data yang di dalamnya merepresentasikan beragam relasi dan juga struktur data yang ada di dalam basis data tersebut.

Adapun esensi dari model basis data merupakan tempat dimana data akan disimpan. Memang kita tidak akan pernah bisa melihat seperti apa model basis data tersebut. Akan tetapi, Anda masih bisa mengeceknya melalui algoritma yang digunkannya.

Secara umum, model basis data dibagi menjadi 4 yakni flat-data, hierarki, jaringan, dan juga relasional. Sementara itu, terdapat model baru yang saat ini telah berhasil dikembangkan oleh beberapa periset yang mana mereka menyebutnya sebagai sebuah sistem pasca relasional. Sementara yang lainnya menggunakan pendekatan yang benar-benar tidak sama. Untuk nama-nama yang saat ini sedang dikembangkan oleh para periset tersebut diantaranya:

-DBMS deduktif -DBMS semantik -DBMS pakar -DBMS berorientasi pada objek -DBMS yang bersifat relasional universal

Perlu dipahami bahwa model basis data merupakan sekumpulan dari berbagai perangkat konseptual yang digunakan untuk menggambarkan suatu data, semantik data, relasi data, batasan data, dan lain sebagainya. Model pada basis data tersebut

nantinya bisa digunakan untuk mengetahui seperti apa pernyataan dari hubungan antar basis data. Model-Model Basis Data :

1. Model Data Flat-le atau Flat-le Data Model:

Model data dari le datar ini bisa juga dinamakan sebagai at le data model. Di sini Anda akan menemukan le yang jumlahnya lebih dari satu dimana kesemua le tersebut bisa dibaca.

Biasanya file tersebut berbentuk teks yang nanti akan disimpan pada suatu fields dimana setiap fieldsnya sudah dilengkapi dengan panjang konstan maupun juga panjang bervariasi yang kemudian dipisahkan oleh masing-masing karakter atau delimeter. Adapula contoh dari flat file data seperti pada gambar dibawah :

1234	5	6789012345678912345	6	789012345678901 23
0123		Muhammad Diar Fadila		Prodi TI-D4
1234		Muhammad Zamzam		Prodi TI-D4
2345		Muhamad Ichsam		Prodi TI-D4
3456		Rizaluardi		Prodi TI-D4
4567		Pertiwi		Prodi TI-D4

Gambar 4.1 Contoh Data Flat

Penjelasannya 1 :

- (a) Seperti yang Anda perhatikan bahwa di sana terdapat 3 elds yang mana setiap elds tersebut mewakili, nama dosen, identifikasi angka, dan juga nama dari program studi atau prodi.
- (b) Setiap elds yang ada sudah mempunyai panjang yang konstan dikarenakan untuk elds sebagai identifikasi angka selalu dimulai dari kolom 1 dan kemudian akan selalu berakhir pada kolom 4
- (c) Sementara itu, elds yang menunjukkan nama dosen akan selalu dimulai dari kolom 6 dan akan selalu berakhir pada kolom 25,dan begitu seterusnya.

Penjelasan ke 2:

Model basis data yang digunakan tersebut memiliki elds dengan panjang yang bervariasi, dimana:

- (a) 0123 merupakan Muhammad diar fadila yang mengambil program studi TI D4
- (b) 1234 yang merupakan Muhammad zamzam yang mengambil program studi Ti-D4
- (c) Sementara untuk 2345 atas nama Muhamad ichsam yang sedang mengambil progam studi Ti-D4

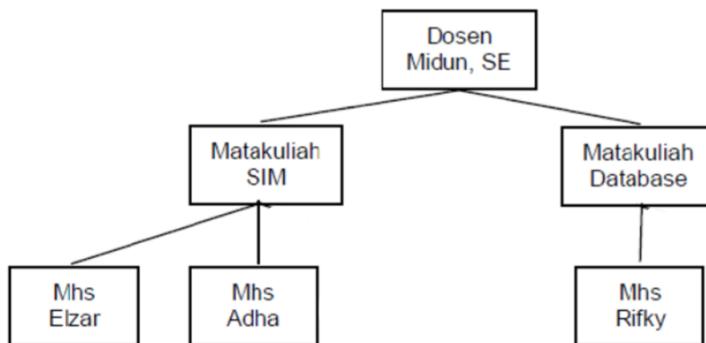
- (d) Dan yang terakhir adalah Pratiwi PS yang mengambil program studi TI-D4
Penjelasan ke 3 :

- i. Model-model yang berasal dari at-le data memiliki panjang yang bervariasi dimana panjangnya tersebut akan dipisahkan oleh sebuah delimiter
- ii. Pada setiap elds akan dipisahkan oleh titik dua jika elds tersebut memiliki panjang yang konstan alias tidak bervariasi.
- iii. Sementara ketika menggunakan elds yang bersifat separator, maka seharusnya elds yang berfungsi sebagai separator tersebut bukanlah merupakan karakter yang ada pada data.

Kelemahan at-le data:

- A. Lokasi sih antara elds data dan juga le harus benar-benar diketahui
- B. Program terlebih dahulu harus dikembangkan agar nanti bisa digunakan untuk mengatur data
- C. Sulit ketika hendak mengatur data secara lebih esien dan juga kurang begitu akurat

4.0.1 Model Data Hirarki



Gambar 4.2 Model data hirarki

Perlu dipahami bahwa model dari data hirarki memiliki satu tingkat diatas data at-le yang berkaitan dengan kemampuan di dalam menemukan maupun juga memelihara setiap relasi yang ada yang dimiliki oleh kelompok data. Beberapa ciri yang dimiliki oleh model basis data jenis ini antara lain sebagai berikut:

- A. Arsitektur yang dimiliki oleh model basis data jenis hirarki tercipta dengan berdasarkan hubungan child atau parent. Pada

saat menggunakan model basis data hirarki,maka nanti root table maupun juga parent table akan berada pada susunan yang paling atas. Kemudian ia akan langsung terhubung ke child table yang dikoneksikan dengan data.

Kelebihan menggunakan model hirarki: Ada beberapa kelebihan dari penggunaan model data hirarki, yaitu:

- B. Data akan dengan cepat bisa dilakukan retrieve
- C. Integritas antar data akan lebih mudah diatur sesuai dengan keperluan

Kelemahan model hirarki: Sementara untuk kelemahanya antara lain :

- A. Seseorang yang menggunakan model ini harus benar-benar familiar terhadap susunan basis data.
- B. Akan terjadi redundansi data

Data Jaringan atau Network Data Model Model data jaringan merupakan model yang sebenarnya merupakan perbaikan dari model hirarki. Adapun perbaikan yang dilakukan adalah dengan penambahan root table ketika hendak melakukan share terhadap child table.Selain itu,disini child table akan memiliki banyak opsi untuk root table. Artinya, ketika ia hendak memperoleh akses menuju child table, maka tidak perlu lagi untuk mengakses root table sebelumnya. Contoh untuk model data jaringan adalah seperti gambar di bawah ini:



Gambar 4.3 Model data jaringan

Kelebihan menggunakan model data jaringan :

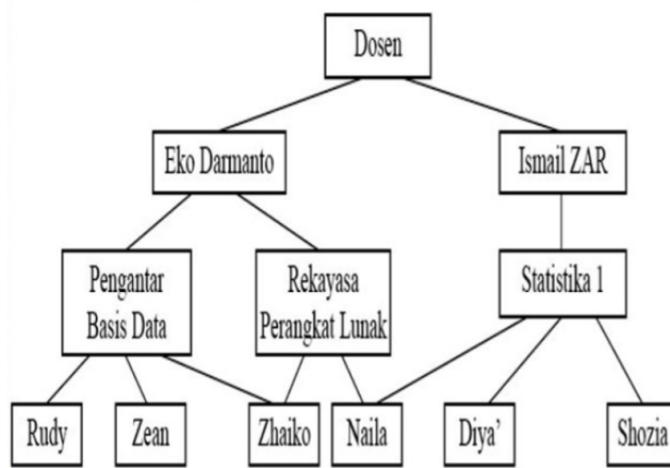
- (a) Data yang mudah diakses
- (b) Kemudahan ketika hendak memodelkan basis data yang bersifat kompleks.
- (c) Bisa dengan mudah ketika hendak membentuk query yang kompleks di dalam retrieve data.

Kekurangan model data jaringan :

- i. Struktur datanya yang tidak mudah ketika hendak melakukan modifikasi
- ii. Pengguna harus benar-benar memahami seperti apa struktur datanya

4.0.2 Model data relasional

Model data yang satu ini merupakan yang paling banyak digunakan. Unit yang disimpan adalah tabel maupun kelompok data yang memiliki hubungan antara yang satu dengan yang lain. Tabel yang saling terhubung tersebut akan dihubungkan oleh sebuah kunci.



Gambar 4.4 Model data relasional

Kelebihan model data relasional :

- i. Kecepatan dalam mengakses data
- ii. Data yang terkenal lebih akurat
- iii. Struktur datanya yang mudah dilakukan modifikasi

- iv. Kemudahan dalam membangun maupun memodifikasi program pada aplikasi

Kekurangan model data relasional :

- i. Pengguna harus benar-benar paham tentang hubungan antar tabel
- ii. Pengguna harus menguasai SQL

Basis Data memiliki beberapa model. Model Basis Data adalah kumpulan dari konsepsipadasuatuBasisDatayangbiasanyaewakilistrukturdanrelasidatayang terdapat pada suatu Basis Data. Sebuah model Basis Data adalah tempat dimana data atau suatu metodologi untuk menyimpan data. Model Basis Data menyatakan hubungan antar rekaman yang tersimpan dalam Basis Data. Jenis-jenis Model Basis Data:

- i. Basis data individual :

Basis data individual merupakan basis data yang telah digunakan oleh perseorangan. Biasanya juga basis data seperti ini banyak kita temui dilingkungan PC. Visual data BASE, Corel Paradox, danFilemaker Pro merupakan contoh perangkat lunak yang biasa akan digunakan untuk mengelolabasi data untuk kepentingan pribadi/kita semua.

- ii. Basis data perusahaan :

Basis data perusahaan merupakan basis data yang dimaksudkan untuk dapat diakses oleh beberapa pegawai dalam sebuah perusahaan dalam sebuah lokasi. Basisdatasepertiinidapatdisimpan dalam sebuah serveratauparapemakaidapat mengakses dari masing-masing komputer yang akan berkedudukan sebagai client.

- iii. Basis data terdistribusi :

Basis data terdistribusi merupakan basis data yang telah disimpan pada sejumlah komputer yang akan terletak pada berbagai lokasi. Model seperti ini banyak digunakan bank yang akan memiliki sejumlah cabang di berbagai kota atau melayani transaksi perbankan yang dapat bersifat online. Berikut ini merupakan jenis - jenis dari basis data

- iv. Basis data publik :

Basis data publik merupakan basis data yang dapat juga diakses oleh siapa saja (publik). Sebagai contohnya, banyak situs web (yang misalnya yahoo atau about.com) yang akan menyediakan data yang bersifat publik dan dapat diambil siapa saja secara gratis. dengan demikian, adakalanya seseorang harus menjadi anggota atau membayar iuran untuk memperoleh data publik.

4.0.3 Fungsi Basis Data

Basis data juga memiliki fungsi umum yang banyak diterapkan dalam dunia industri di seluruh dunia untuk dapat mendukung sistem atau aplikasi yang digunakan pada setiap industry. Berikut beberapa ini fungsi dasar dari database yakni :

- i. Database berfungsi untuk mengklasifikasikan data untuk mudah digunakan dan dipahami penggunanya.
- ii. Menghindari adanya duplikasi dan juga inkonsistensi dari suatu data.
- iii. Memudahkan dalam menyimpan dan mengakses data.
- iv. Memudahkan dalam mengupdate dan menghapus data.
- v. Menjamin kualitas data dan informasi yang terkandung didalamnya dan dapat diakses.
- vi. Merupakan suatu solusi penyimpanan data.
- vii. Mendukung kinerja aplikasi dalam penyimpanan data.

4.0.4 Manfaat Basis Data

- i. Kecepatan dan Kemudahan (Speed)
- ii. Kebersamaan Pemakaian (Sharability)
- iii. Pemusatan Kontrol Data
- iv. Esiensi Ruang Penyimpanan (Space)
- v. Keakuratan (Accuracy)
- vi. Ketersediaan (Availability)
- vii. Kelengkapan (Completeness)
- viii. Keamanan (Security)
- ix. Kemudahan dalam Pembuatan Program Aplikasi Baru

4.0.5 Komponen Basis Data

Basis data merupakan sistem yang terdiri dari kumpulan file atau tabel yang saling berhubungan dan Database Management System yang memungkinkan ada beberapa pemakai untuk dapat mengakses atau manipulasi file tersebut . Dalam Sistem Basis data juga dapat memiliki beberapa komponen yakni:

2. Perangkat Keras (Hardware):

Perangkat keras yang juga biasanya terdapat dalam sistem basis data adalah memori sekunder hardisk.

3. Sistem Operasi (Operating System) : Sistem Operasi (Operating System) adalah program yang dapat mengaktifkan maupun mengfungksikan sistem komputer, mengendalikan seluruh sumber daya (resource) atau melakukan operasi-operasi dalam komputer. Sistem Operasi yang banyak digunakan seperti: MS-DOS, MS-Windows 95 MS Windows NT, atau Unix.
4. Basis data (Database): Sebuah basis data merupakan dapat juga memiliki beberapa basis data. Setiap basis data dapat berisi maupun akan memiliki sejumlah objek basis data seperti le atau tabel Database.
5. Management System (DBMS):
Pengolahan basis data secara sik tidak dilakukan oleh pemakai secara langsung, tetapi juga ditangani oleh sebuah perangkat lunak yang disebut dengan DBMS yang menentukan bagaimana data disimpan, diubah atau diambil kembali.

6. Pemakai (User):
Bagi pemakai dapat berinteraksi dengan basis data atau juga akan memanipulasi data dalam program yang ditulis dalam bahasa pemrograman

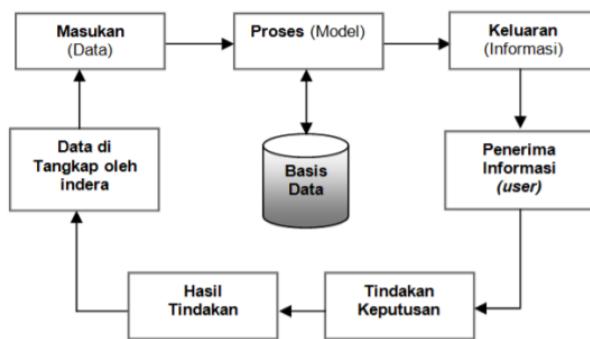
BAB 5

RANCANGAN BASIS DATA

Perancangan Basis Data adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Istilah data bermakna untuk mengetahui fakta-fakta yang dapat direkam dan disimpan pada media komputer. Definisi ini kini berkembang untuk mencerminkan realitas baru. Basis data sekarang digunakan untuk menyimpan objek seperti dokumen, foto, suara, dan video, sebagai tambahan dari data teks dan data numerik. Untuk mencerminkan realitas, kita menggunakan definisi yang diperluas berikut: Data terdiri dari fakta-fakta, hasil-hasil pengujian, grafik, gambar, dan video yang mempunyai arti dalam lingkungan pengguna .Kita telah mendefinisikan basis data sebagai kumpulan data yang terorganisasi dan saling berhubungan. Terorganisasi maksudnya adalah data yang terstruktur sehingga mudah disimpan, dimanipulasi, dan diambil kembali oleh pengguna.Saling berhubungan maksudnya adalah data menggambarkan suatu domain yang menjadi perhatian sekelompok pengguna dan pengguna pengguna dapat menggunakan data untuk menjawab pertanyaan yang menjadi perhatian dari domain tersebut.

5.0.1 Siklus Informasi

Data dan informasi akan saling berkesinambungan sehingga membentuk suatu siklus yang disebut information cycle (siklus informasi). Data ditangkap oleh indera ke-mudian menjadi inputan dalam sebuah model untuk diubah menjadi informasi bagi penerimanya yang nantinya akan membantu pengambilan keputusan dan menjadi sebuah hasil tindakan.



Gambar 5.1 Siklus Informasi

Pentingnya data dan informasi

1. Data dan informasi sebagai sebuah aset penting perusahaan/organisasi.
2. Informasi yang benar dapat menjadikan suatu perusahaan/organisasi memperoleh margin untuk melakukan aksi.
3. Data dan informasi sebagai salah satu parameter kemajuan perusahaan/organisasi (maturity level).

5.0.2 Sistem Basis Data dan Sistem File

Pada sebuah institusi, data merupakan salah satu hal yang sangat penting. Setiap bagian/divisi dari institusi memiliki data sendiri-sendiri. Tapi setiap bagian pun membutuhkan sebagian data dari bagian yang lain. Hal ini yang biasa dikenal sebagai shared data. Setiap divisi memiliki aplikasi sendiri-sendiri dalam melakukan manipulasi dan pengambilan data tersebut. Setiap aplikasi memiliki file-file dalam sistem operasi yang digunakan untuk menyimpan data-data. Seiring dengan berkembangnya institusi, bertambahnya bagian/divisi, bertambah pula data dan aplikasi yang digunakan. Bertambahnya aplikasi, bertambah pula file-file yang dibuat. Gaya sistem pemrosesan-file tersebut menyebabkan setiap data disimpan dalam bentuk record dalam berbagai macam file, dan diperlukan aplikasi yang berbeda dalam melakukan pengambilan record dari, dan penambahan record ke dalam file. Hal ini berlaku pada masa sebelum adanya Sistem Basis Data (DBMS).

Menyimpan data dalam bentuk file yang berbeda-beda, memiliki kekurangan-kekurangan:

1. Data redundancy dan inconsistency.

Dikarenakan programer yang berbeda membuat file dan aplikasi masing-masing, menyebabkan beragam format dan aplikasi yang dibuat. Bahkan, aplikasi pun dibuat menggunakan bahasa pemrograman yang berbeda-beda. Lebih jauh lagi, data atau informasi yang sama bisa terdapat dalam beberapa file yang berbeda. Ini yang disebut dengan redundancy. Redundancy data ini lama kelamaan akan menyebabkan inconsistency dari data.

2. Kesulitan dalam pengaksesan data.

Dikarenakan setiap aplikasi memiliki file tersendiri untuk penyimpanan dan pengambilan data, maka jika suatu bagian dari institusi membutuhkan data dari bagian lain, akan menemui kesulitan. Hal ini dikarenakan aplikasi yang dimiliki bagian tersebut, tidak dapat membaca file yang terdapat di bagian lain

3. Isolasi data

Dikarenakan data tersebar dalam berbagai macam file, dan file tersebut dalam beragam format, pembuatan aplikasi baru akan terasa sulit ketika harus membaca format dari masing-masing file tersebut.

4. Masalah integritas

Data yang disimpan harus memenuhi hal yang dinamakan dengan consistency constraint. Jika sebuah constraint berubah, maka seluruh aplikasi yang digunakan harus mengakomodasinya. Masalah akan muncul, jika constraint melibatkan beberapa data dari file yang berbeda-beda.

5. Masalah keamanan

Tidak semua pengguna dari basis data dapat mengakses semua data. Hal ini akan sulit dilakukan jika menggunakan gaya penyimpanan data dalam file.

5.0.3 Basis Data dan Sistem Basis Data (DBMS)

Basis data adalah penyimpanan kumpulan informasi secara sistematis dalam sebuah komputer sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat Lunak yang digunakan untuk mengelola dan memanggil query basis data disebut Sistem Manajemen Basis Data (Database Management System, DBMS). DBMS memiliki karakteristik sebagai berikut:

- (a) Software program
- (b) Supplements operating sistem
- (c) Manages data
- (d) Queries data and generates reports
- (e) Data security

Sedangkan sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses pekerjaan. Sehingga bisa dikatakan bahwa sistem basis data adalah sistem yang terdiri atas kumpulan file-file yang saling berhubungan dan dikelola oleh program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain yang memiliki otoritas untuk mengakses dan memanipulasi data tersebut. Kelebihan pemakaian DBMS adalah:

- (a) Data berdiri sendiri (Data Independence)
- (b) Pengaksesan data efisien (Efficient data access)
- (c) Integritas data dan keamanan terjamin (Data integrity and security)
- (d) Administrasi data (Data administration)
- (e) Dapat diakses bersamaan (Concurrent access)
- (f) Recovery saat terjadi kegagalan (Crash recovery)
- (g) Mengurangi waktu pembangunan aplikasi (Reduced application development time)

5.0.4 Komponen Sistem Basis Data

Komponen-komponen pada sebuah sistem basis data antara lain:

- (a) Perangkat keras

- (b) Sistem operasi
- (c) Basis data
- (d) DBMS (Database Management System)
- (e) Pemakai
- (f) Aplikasi lain

5.0.5 Abstraksi Data

Tujuan utama dari sistem basis data adalah untuk menyediakan fasilitas untuk view data secara abstrak bagi penggunanya. Namun bagaimana sistem menyimpan dan mengelola data tersebut, hanya diketahui oleh sistem itu sendiri. Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data. Berikut ini tiga level abstraksi data:

- (a) Level fisik
 - Merupakan level terendah pada abstraksi data yang menunjukkan bagaimana sesungguhnya data disimpan. Pada level ini pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.
- (b) Level logik
 - Merupakan level berikutnya pada abstraksi data, menggambarkan data apa yang disimpan pada basis data dan hubungan apa saja yang ada di antara data tersebut.
- (c) Level view
 - Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data. Banyak user dalam sistem basis data tidak akan terlibat dengan semua data atau informasi yang ada atau yang disimpan. Para user umumnya hanya membutuhkan sebagian data atau informasi dalam basis data yang kemunculannya di mata user diatur oleh aplikasi end user.

5.0.6 Model Basis Data

- (a) Hierarchical
 - Memiliki struktur pohon dimana field hanya memiliki satu buah induk (parent), masingmasing parent memiliki banyak child (anak). Model ini memiliki kecepatan yang baik.
- (b) Network

Relationship dibuat menggunakan linked list (pointer). Berbeda dengan model hierarchical satu anak dapat memiliki beberapa induk. Model ini memiliki fleksibilitas yang tinggi.

(c) Relational

Model ini direpresentasikan dalam tabel dua dimensi, tabel-tabel tersebut memiliki hubungan yang disebut dengan relasi. Model ini memiliki fleksibilitas dan kecepatan yang tinggi.

(d) Object oriented

Object Oriented Database adalah sebuah sistem database yang menggabungkan semua konsep object oriented seperti pewarisan, abstraksi, enkapsulasi, dll. Model ini dapat berinteraksi dengan baik dengan bahasa pemrograman berorientasi objek seperti java dan C++.

5.0.7 Perancangan Database

Di dalam suatu organisasi yang besar, sistem database merupakan bagian penting pada sistem informasi, karena di perlukan untuk mengelola sumber informasi pada organisasi tersebut. Untuk mengelola sumber informasi tersebut yang pertama kali di lakukan adalah merancang suatu sistem database agar informasi yang ada pada organisasi tersebut dapat digunakan secara maksimal.

Tujuan Perancangan Database :

- (a) Untuk memenuhi kebutuhan akan informasi dari pengguna dan aplikasi
- (b) Menyediakan struktur informasi yang natural dan mudah di mengerti oleh pengguna
- (c) Mendukung kebutuhan pemrosesan dan beberapa obyek kinerja dari suatu sistem database

Berikut ini siklus kehidupan sistem informasi di mana terdapat siklus kehidupan sistem database

5.0.7.1 Siklus Kehidupan Sistem Informasi (Macro Life Cycle) Tahapan yang ada pada siklus kehidupan sistem informasi yaitu :

i. Analisa Kelayakan

Tahapan ini memfokuskan pada penganalisaan areal aplikasi yang unggul , mengidentifikasi pengumpulan informasi dan penyebarannya, mempelajari keuntungan dan kerugian , penentuan kompleksitas data dan proses, dan menentukan prioritas aplikasi yang akan digunakan.

ii. Analisa dan Pengumpulan Kebutuhan Pengguna

Kebutuhankebutuhan yang detail dikumpulkan dengan berinteraksi pada sekelompok pemakai atau pemakai individu. Mengidentifikasi masalah yang ada dan kebutuhan-butuhan, ketergantungan antar aplikasi, komunikasi dan prosedur laporan.

iii. Perancangan

Perancangan terbagi menjadi dua yaitu : perancangan sistem database dan sistem aplikasi

iv. Implementasi Mengimplementasikan sistem informasi dengan database yang ada

v. Pengujian dan Validasi

Pengujian dan validasi sistem database dengan kriteria kinerja yang diinginkan oleh pengguna.

vi. Pengoperasian dan Perawatan

Pengoperasian sistem setelah di validasi disertai dengan pengawasan dan perawatan sistem

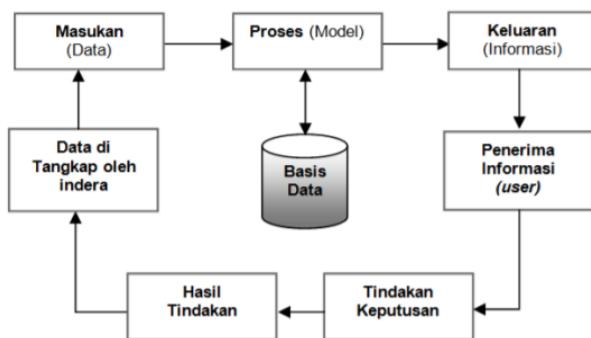
BAB 6

RANCANGAN BASIS DATA

Perancangan Basis Data adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Istilah data bermakna untuk mengetahui fakta-fakta yang dapat direkam dan disimpan pada media komputer. Definisi ini kini berkembang untuk mencerminkan realitas baru. Basis data sekarang digunakan untuk menyimpan objek seperti dokumen, foto, suara, dan video, sebagai tambahan dari data teks dan data numerik. Untuk mencerminkan realitas, kita menggunakan definisi yang diperluas berikut: Data terdiri dari fakta-fakta, hasil-hasil pengujian, grafik, gambar, dan video yang mempunyai arti dalam lingkungan pengguna .Kita telah mendefinisikan basis data sebagai kumpulan data yang terorganisasi dan saling berhubungan. Terorganisasi maksudnya adalah data yang terstruktur sehingga mudah disimpan, dimanipulasi, dan diambil kembali oleh pengguna.. Saling berhubungan maksudnya adalah data menggambarkan suatu domain yang menjadi perhatian sekelompok pengguna dan pengguna pengguna dapat menggunakan data untuk menjawab pertanyaan yang menjadi perhatian dari domain tersebut.

6.0.1 Siklus Informasi

Data dan informasi akan saling berkesinambungan sehingga membentuk suatu siklus yang disebut information cycle (siklus informasi). Data ditangkap oleh indera ke-mudian menjadi inputan dalam sebuah model untuk diubah menjadi informasi bagi penerimanya yang nantinya akan membantu pengambilan keputusan dan menjadi sebuah hasil tindakan.



Gambar 6.1 *Siklus Informasi*

Pentingnya data dan informasi

1. Data dan informasi sebagai sebuah aset penting perusahaan/organisasi.
2. Informasi yang benar dapat menjadikan suatu perusahaan/organisasi memperoleh margin untuk melakukan aksi.
3. Data dan informasi sebagai salah satu parameter kemajuan perusahaan/organisasi (maturity level).

6.0.2 Sistem Basis Data dan Sistem File

Pada sebuah institusi, data merupakan salah satu hal yang sangat penting. Setiap bagian/divisi dari institusi memiliki data sendiri-sendiri. Tapi setiap bagian pun membutuhkan sebagian data dari bagian yang lain. Hal ini yang biasa dikenal sebagai shared data. Setiap divisi memiliki aplikasi sendiri-sendiri dalam melakukan manipulasi dan pengambilan data tersebut. Setiap aplikasi memiliki file-file dalam sistem operasi yang digunakan untuk menyimpan data-data. Seiring dengan berkembangnya institusi, bertambahnya bagian/divisi, bertambah pula data dan aplikasi yang digunakan. Bertambahnya aplikasi, bertambah pula file-file yang dibuat. Gaya sistem pemrosesan-file tersebut menyebabkan setiap data disimpan dalam bentuk record dalam berbagai macam file, dan diperlukan aplikasi yang berbeda dalam melakukan pengambilan record dari, dan penambahan record ke dalam file. Hal ini berlaku pada masa sebelum adanya Sistem Basis Data (DBMS).

Menyimpan data dalam bentuk file yang berbeda-beda, memiliki kekurangan-kekurangan:

1. Data redundancy dan inconsistency.

Dikarenakan programer yang berbeda membuat file dan aplikasi masing-masing, menyebabkan beragam format dan aplikasi yang dibuat. Bahkan, aplikasi pun dibuat menggunakan bahasa pemrograman yang berbeda-beda. Lebih jauh lagi, data atau informasi yang sama bisa terdapat dalam beberapa file yang berbeda. Ini yang disebut dengan redundancy. Redundancy data ini lama kelamaan akan menyebabkan inconsistency dari data.

2. Kesulitan dalam pengaksesan data.

Dikarenakan setiap aplikasi memiliki file tersendiri untuk penyimpanan dan pengambilan data, maka jika suatu bagian dari institusi membutuhkan data dari bagian lain, akan menemui kesulitan. Hal ini dikarenakan aplikasi yang dimiliki bagian tersebut, tidak dapat membaca file yang terdapat di bagian lain

3. Isolasi data

Dikarenakan data tersebar dalam berbagai macam file, dan file tersebut dalam beragam format, pembuatan aplikasi baru akan terasa sulit ketika harus membaca format dari masing-masing file tersebut.

4. Masalah integritas

Data yang disimpan harus memenuhi hal yang dinamakan dengan consistency constraint. Jika sebuah constraint berubah, maka seluruh aplikasi yang digunakan harus mengakomodasinya. Masalah akan muncul, jika constraint melibatkan beberapa data dari file yang berbeda-beda.

5. Masalah keamanan

Tidak semua pengguna dari basis data dapat mengakses semua data. Hal ini akan sulit dilakukan jika menggunakan gaya penyimpanan data dalam file.

6.0.3 Basis Data dan Sistem Basis Data (DBMS)

Basis data adalah penyimpanan kumpulan informasi secara sistematis dalam sebuah komputer sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat Lunak yang digunakan untuk mengelola dan memanggil query basis data disebut Sistem Manajemen Basis Data (Database Management System, DBMS). DBMS memiliki karakteristik sebagai berikut:

- (a) Software program
- (b) Supplements operating sistem
- (c) Manages data
- (d) Queries data and generates reports
- (e) Data security

Sedangkan sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses pekerjaan. Sehingga bisa dikatakan bahwa sistem basis data adalah sistem yang terdiri atas kumpulan file-file yang saling berhubungan dan dikelola oleh program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain yang memiliki otoritas untuk mengakses dan memanipulasi data tersebut. Kelebihan pemakaian DBMS adalah:

- (a) Data berdiri sendiri (Data Independence)
- (b) Pengaksesan data efisien (Efficient data access)
- (c) Integritas data dan keamanan terjamin (Data integrity and security)
- (d) Administrasi data (Data administration)
- (e) Dapat diakses bersamaan (Concurrent access)
- (f) Recovery saat terjadi kegagalan (Crash recovery)
- (g) Mengurangi waktu pembangunan aplikasi (Reduced application development time)

6.0.4 Komponen Sistem Basis Data

Komponen-komponen pada sebuah sistem basis data antara lain:

- (a) Perangkat keras
- (b) Sistem operasi
- (c) Basis data
- (d) DBMS (Database Management System)
- (e) Pemakai
- (f) Aplikasi lain

6.0.5 Abstraksi Data

Tujuan utama dari sistem basis data adalah untuk menyediakan fasilitas untuk view data secara abstrak bagi penggunanya. Namun bagaimana sistem menyimpan dan mengelola data tersebut, hanya diketahui oleh sistem itu sendiri. Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data. Berikut ini tiga level abstraksi data:

- (a) Level fisik
Merupakan level terendah pada abstraksi data yang menunjukkan bagaimana sesungguhnya data disimpan. Pada level ini pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.
- (b) Level lojik
Merupakan level berikutnya pada abstraksi data, menggambarkan data apa yang disimpan pada basis data dan hubungan apa saja yang ada di antara data tersebut.
- (c) Level view
Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data. Banyak user dalam sistem basis data tidak akan terlibat dengan semua data atau informasi yang ada atau yang disimpan. Para user umumnya hanya membutuhkan sebagian data atau informasi dalam basis data yang kemunculannya di mata user diatur oleh aplikasi end user.

6.0.6 Model Basis Data

(a) Hierarchical

Memiliki struktur pohon dimana field hanya memiliki satu buah induk (parent), masingmasing parent memiliki banyak child (anak). Model ini memiliki kecepatan yang baik.

(b) Network

Relationship dibuat menggunakan linked list (pointer). Berbeda dengan model hierarchical satu anak dapat memiliki beberapa induk. Model ini memiliki fleksibilitas yang tinggi.

(c) Relational

Model ini direpresentasikan dalam tabel dua dimensi, tabel-tabel tersebut memiliki hubungan yang disebut dengan relasi. Model ini memiliki fleksibilitas dan kecepatan yang tinggi.

(d) Object oriented

Object Oriented Database adalah sebuah sistem database yang menggabungkan semua konsep object oriented seperti pewarisan, abstraksi, enkapsulasi, dll. Model ini dapat berinteraksi dengan baik dengan bahasa pemrograman berorientasi objek seperti java dan C++.

6.0.7 Perancangan Database

Di dalam suatu organisasi yang besar, sistem database merupakan bagian penting pada sistem informasi, karena di perlukan untuk mengelola sumber informasi pada organisasi tersebut. Untuk mengelola sumber informasi tersebut yang pertama kali di lakukan adalah merancang suatu sistem database agar informasi yang ada pada organisasi tersebut dapat digunakan secara maksimal.

Tujuan Perancangan Database :

- (a) Untuk memenuhi kebutuhan akan informasi dari pengguna dan aplikasi
- (b) Menyediakan struktur informasi yang natural dan mudah di mengerti oleh pengguna
- (c) Mendukung kebutuhan pemrosesan dan beberapa obyek kinerja dari suatu sistem database

Berikut ini siklus kehidupan sistem informasi di mana terdapat siklus kehidupan sistem database

6.0.7.1 Siklus Kehidupan Sistem Informasi (Macro Life Cycle) Tahapan yang ada pada siklus kehidupan sistem informasi yaitu :

i. Analisa Kelayakan

Tahapan ini memfokuskan pada penganalisaan areal aplikasi yang ungul , mengidentifikasi pengumpulan informasi dan penyebarannya, mempelajari keuntungan dan kerugian , penentuan kompleksitas data dan proses, dan menentukan prioritas aplikasi yang akan digunakan.

ii. Analisa dan Pengumpulan Kebutuhan Pengguna

Kebutuhankebutuhan yang detail dikumpulkan dengan berinteraksi pada sekelompok pemakai atau pemakai individu. Mengidentifikasikan masalah yang ada dan kebutuhan-butuhan, ketergantungan antar aplikasi, komunikasi dan prosedur laporan.

iii. Perancangan

Perancangan terbagi menjadi dua yaitu : perancangan sistem database dan sistem aplikasi

iv. Implementasi Mengimplementasikan sistem informasi dengan database yang ada

v. Pengujian dan Validasi

Pengujian dan validasi sistem database dengan kriteria kinerja yang diinginkan oleh pengguna.

vi. Pengoperasian dan Perawatan

Pengoperasian sistem setelah di validasi disertai dengan pengawasan dan perawatan sistem

BAB 7

DATABASE SERVER

Database server adalah program komputer yang menyediakan layanan data lainnya ke komputer atau program komputer, seperti yang ditetapkan oleh model klien-server. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program server database. Database sistem manajemen database yang sering menyediakan fungsi server, dan beberapa DBMSs (misalnya, MySQL) secara eksklusif bergantung pada model klien-server untuk akses data. Database server memiliki fungsi seperti semua data untuk organisasi dapat disimpan di satu lokasi. Database server menambahkan tingkat keamanan data. Database server menyediakan layanan database management service dimana data disusun dengan cara tertentu sehingga meningkatkan pencarian dan pengambilan data. Beberapa client dapat mengakses data yang disimpan di database server dalam satu waktu tanpa saling mengganggu satu sama lain.

Database server memiliki sistematika kerjanya adalah Client-server model dapat diartikan sebagai model dari suatu sistem yang membagi proses sistem antara server yang mengolah database dan client yang menjalankan aplikasi. Database server mengurangi beban akses data oleh client pada server. Database dapat diakses oleh beberapa client secara bersamaan dimana data yang diakses hanya atau diubah berasal dari satu sumber yaitu database pada server.

Server tersebut diakses baik melalui suatu "front end" yang berjalan di komputer pengguna yang menampilkan data yang diminta atau "back end" yang berjalan pada server dan menangani tugas-tugas seperti analisis data dan penyimpanan.

Dalam model master-slave, database server master adalah lokasi pusat dan utama data sementara database server budak disinkronisasi backup dari master bertindak sebagai proxy. Beberapa contoh dari server basis data Oracle, DB2, Informix, Ingres, SQL Server. Setiap server menggunakan query sendiri logika dan struktur. Bahasa query SQL kurang lebih sama di semua server database.

Jenis-jenis software program Database Server:

1. MySQL



Gambar 7.1 *Mysql*

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : data management system) atau DMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan cipta untuk code sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : David Axmark, Allan Larsson, dan Michael Widenius.

Adapula kelebihan dari penggunaan SQL antara lain :

- (a) Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, dan masih banyak lagi.
- (b) Free (bebas didownload) MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

- (c) stabil dan tangguh, fleksibel dengan berbagai pemrograman
- (d) Security yang baik mendukung transaksi
- (e) dukungan dari banyak komunitas perkembangan software yang cukup cepat
- (f) kemudahan management database

2. ORACLE



Gambar 7.2 *Oracle*

Oracle adalah relational database management system (RDBMS) untuk men-gelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle Server menyediakan solusi yang efisien dan efektif karena kemampuannya dalam hal sebagai berikut: Dapat bekerja di lingkungan client/server (pemrosesan terse-bar) Menangani manajemen space dan basis data yang besar Mendukung akses data secara simultan Performansi pemrosesan transaksi yang tinggi Menjamin ketersediaan yang terkontrol Lingkungan yang terreplikasi Database merupakan salah satu komponen dalam teknologi informasi yang mutlak diperlukan oleh semua organisasi yang ingin mempunyai suatu sistem informasi yang terpadu untuk menunjang kegiatan organisasi demi mencapai tujuannya. Karena pentingnya peran database dalam sistem informasi, tidaklah mengherankan bahwa terdapat banyak pilihan software Database Management System (DBMS) dari berbagai vendor baik yang gratis maupun yang komersial. Beberapa contoh DBMS yang populer adalah MySQL, MS SQL Server, Oracle, IBM DB/2, dan PostgreSQL.

Selain itu adapula kelebihan dan kekurangan dari oracle ini yaitu :

Fleksibilitas adalah kemampuan untuk menyesuaikan diri dengan berbagai kebutuhan dan kondisi khusus yang dapat berubah-ubah. Sebagai contoh, organisasi yang besar membutuhkan server yang terdistribusi dan memiliki redundancy sehingga pelayanan bisa diberikan secara cepat dan tidak terganggu jika ada server yang mati. Organisasi tersebut juga mempunyai berbagai macam aplikasi yang dibuat dengan beragam bahasa pemrograman dan berjalan di berbagai platform yang berbeda. Oracle memiliki banyak sekali fitur yang dapat memenuhi tuntutan fleksibilitas dari organisasi besar tersebut. Berbagai fitur tersebut membuat Oracle menjadi DBMS yang rumit dan sulit untuk dipelajari, namun itu adalah harga yang harus dibayar untuk mendapatkan fleksibilitas yang dibutuhkan dalam sistem informasi di organisasi yang berukuran besar. Skalabilitas mengacu pada kemampuan untuk terus berkembang dengan penambahan sumber daya. Organisasi yang besar harus mampu melakukan transaksi

data dalam volume yang besar dan akan terus bertambah besar. Jika dijalankan hanya pada satu server saja, MySQL memang bisa berjalan lebih cepat daripada Oracle. Namun jika satu server sudah tidak bisa lagi menangani beban yang terus bertambah besar, kinerja MySQL mengalami stagnasi karena keterbatasan server tersebut. Namun Oracle mendukung fitur Grid yang dapat mendayagunakan lebih dari satu server serta data storage dengan mudah dan transparan. Hanya dengan menambahkan server atau data storage ke dalam Oracle Grid, maka kinerja dan kapasitas Oracle dapat terus berkembang untuk mengikuti beban kerja yang terus meningkat. Demikianlah salah satu (atau dua) keunggulan dari Oracle. Tidaklah mengherankan bahwa meskipun Oracle merupakan DBMS yang paling rumit dan paling mahal di dunia, perusahaan-perusahaan besar memilih Oracle dan tidak menggunakan DBMS seperti MySQL yang gratis karena mereka membutuhkan fleksibilitas dan skalabilitas dalam sistem informasi yang mereka gunakan.

3. FIREBIRD



Gambar 7.3 *Firebird*

Firebird adalah salah satu aplikasi RDBMS (Relational Database Management System) yang bersifat open source. Awalnya adalah perusahaan Borland yang sekitar tahun 2000 mengeluarkan versi beta dari aplikasi database-nya InterBase 6.0 dengan sifat open source. Namun entah kenapa tiba-tiba Borland tidak lagi mengeluarkan versi InterBase secara open source, justru kembali ke pola komersial software. Tapi pada saat yang bersamaan programmer-progammer yang tertarik dengan source code InterBase 6.0 tersebut lalu membuat suatu team untuk mengembangkan source code database ini dan kemudian akhirnya diberinama Firebird.

Firebird (juga disebut FirebirdSQL) adalah sistem manajemen basisdata relasional yang menawarkan fitur-fitur yang terdapat dalam standar ANSI SQL-99 dan SQL-2003. RDBMS ini berjalan baik di Linux, Windows, maupun pada sejumlah platform Unix. Firebird ini diarahkan dan di-maintain oleh Firebird-SQL Foundation. Ia merupakan turunan dari Interbase versi open source milik Borland. Karena itulah Interbase dan Firebird sebenarnya mempunyai CORE yang sama karena awalnya sama dikembangkan oleh Borland.

Selain itu ada kemampuan dan kelebihan dari firebird ini sendiri

Beberapa kemampuan dari open source DBMS ini antara lain: - Firebird support dengan transaksi layaknya pada database komersial lainnya. Sebuah transaksi bisa di-commit atau di-rollback dengan mudah. Bahkan Firebird support dengan savepoint pada suatu transaksi dan kita bisa melakukan rollback kembali ke savepoint yang kita tentukan tadi (ini mirip seperti fasilitas pada Oracle).

-Firebird menggunakan sintaks standard untuk menciptakan suatu foreign key - Firebird support row level locks, secara default Firebird menggunakan apa yang disebut dengan multi-version concurrency system. Ini artinya bahwa semua session pada database akan melihat data yang lama sampai data yang baru sudah di-commit ke dalam database. Sebagai alternative untuk locking juga bisa digunakan perintah select for update with lock -Firebird support stored procedure dan triggers dengan bahasa yang standard sehingga tidak akan membingungkan bagi Anda yang ingin belajar. Triggers pada Firebird mirip seperti yang dijumpai dalam Oracle yaitu menggunakan before atau after insert, update atau delete. Ini berbeda dengan sistem trigger pada Sybase atau MS SQL Server yang menggunakan tabel virtual inserted dan deleted. -Firebird bisa melakukan replikasi, solusi untuk replikasi kebanyakan dibuat oleh pihak ketiga, tetapi sebenarnya teknik replikasi ini seperti konsep trigger yang selalu memonitor adanya operasi insert, update atau delete ke dalam database. -Firebird support dengan multiple data file, ya, Firebird bisa menggunakan lebih dari satu file sebagai single logic database. Ini sangat berguna bagi para DBA (Database Administrator) untuk mengadministrasi database. -Software untuk mengadministrasi mudah didapat karena banyak sekali software untuk mengadministrasi database Firebird, misalnya saja EMS IB Manager, IBConsole, isql, FB-Manager, Marathon dan banyak lagi yang lainnya. Aplikasi tersebut ada yang komersial atau bahkan ada yang open source. -Library connection untuk Firebird yang sudah tersedia ada banyak, ada driver untuk ODBC, JDBC bahkan .NET database provider. Bahkan dari PHP juga sudah disediakan library untuk koneksi ke database Firebird ini. -Banyaknya fasilitas support dan maintenance karena banyaknya komunitas Firebird. Di Indonesia sendiri komunitas yang sering menggunakan Firebird biasanya adalah programmer Delphi.

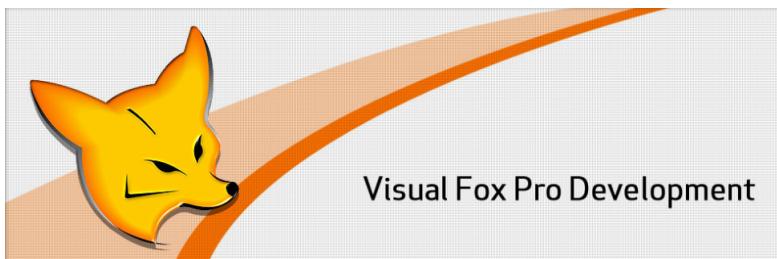
4. MICROSOFT SQL SERVER 2000



Gambar 7.4 *Sql server*

Microsoft SQL Server 2000 adalah perangkat lunak relational database management system (RDBMS) yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas. Microsoft SQL Server 2000 merupakan produk andalan Microsoft untuk database server. Kemampuannya dalam manajemen data dan kemudahan dalam pengoperasiannya membuat RDBMS ini menjadi pilihan para database administrator. DBMS merupakan suatu sistem perangkat lunak untuk memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan manipulasi data yang ada. Sedangkan RDBMS atau Relationship Database Management System merupakan salah satu jenis DBMS yang mendukung adanya relationship atau hubungan antar table. RDBMS (Relational Database Management System) adalah perangkat lunak untuk membuat dan mengelola database, sering juga disebut sebagai database engine. Istilah RDBMS, database server-software, dan database engine mengacu ke hal yang sama; sedangkan RDBMS bukanlah database. Beberapa contoh dari RDBMS diantaranya Oracle, Ms SQL Server, MySQL, DB2, Ms Access.

5. VISUAL FOXPRO 6.0



Gambar 7.5 *Visual foxpro*

Pada tahun 1984, Fox Software memperkenalkan FoxBase untuk menyaingi dBase II Ashton-Tate. Pada saat itu FoxBase hanyalah perangkat lunak kecil yang berisi bahasa pemrograman dan mesin pengolah data. FoxPro memperkenalkan GUI (Graphical Unit Interface) pada tahun 1989. FoxPro berkembang menjadi Visul FoxPro pada tahun 1995. kemampuan pemrograman prosural tetap dipertahankan dan dilengkapi dengan pemrograman berorientasi objek. Visual FoxPro 6.0 dilengkapi dengan kemampuan untuk berinteraksi dengan produk desktop dan client/server lain dan juga dapat membangun aplikasi yang berbasis Web. Dengan adanya Visual Studio, FoxPro menjadi anggotanya. Sasaran utama Visual Studio adalah menyediakan alat bantu pemrograman dan database untuk mengembangkan perangkat lunak yang memenuhi tuntutan zaman. Model data yang digunakan Visual FoxPro yaitu model relasional. Model Relasional merupakan model yang paling sederhana sehingga mudah dimengerti oleh pengguna, serta merupakan model yang paling popular saat ini. Model ini menggunakan sekumpulan table berdimensi dua (yang disebut relasi atau table), dengan masing-masing relasi tersusun atas tupel atau baris dan atribut. Relasi dirancang sedemikian rupa sehingga dapat menghilangkan kemubajiran data dan menggunakan kunci tamu untuk berhubungan dengan relasi lain.

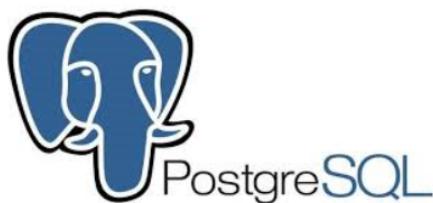
6. DATABASE DEKSTOP PARADOX



Gambar 7.6 *paradox*

Database desktop merupakan suatu program Add-Ins, yaitu program terpisah yang langsung terdapat pada Borland Delphi. Pada database desktop terdapat beberapa DBMS yang terintegrasi di dalamnya antara lain Paradox 7, Paradox 4, Visual dBBase, Foxpro, Ms. SQL, Oracle, Ms. Acces, db2 dan interbase. Dari beberapa DBMS tersebut kita akan memilih salah satu yaitu Paradox yang akan dibahas lebih lanjut, khususnya Paradox 7. Dalam Paradox 7 ini, pada 1 file database hanya mengizinkan 1 tabel, berbeda dengan DBMS lain yang mengizinkan beberapa tabel pada 1 file database seperti pada Ms. Acces.

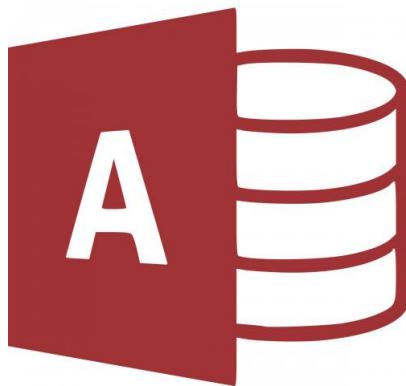
7. POSTGRE SQL



Gambar 7.7 *postgre*

PostgreSQL merupakan tool untuk membuat database server yang bersifat open source yang memiliki lisensi GPL (General Public License). PostgreSQL mendukung bahasa pemrograman seperti : SQL, C, C++, Java, PHP dan lainnya. Dengan lisensi GPL, PostgreSQL dapat digunakan, dimodifikasi dan didistribusikan oleh setiap orang tanpa perlu membayar lisensi (free of charge) baik untuk keperluan pribadi, pendidikan maupun komersil. Kebanyakan PostgreSQL tersedia untuk pengguna Linux.

8. MICROSOFT ACCESS



Gambar 7.8 *Microsoft Access*

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007. Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/programmer yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

BAB 8

STUDI KASUS

8.1 Oracle Apex

Pada proses kali ini ada studi kasus dalam penggunaan aplikasi oracle apex. Dalam studi kasus membuat aplikasi pengolahan data administrasi barang menggunakan aplikasi apex online.

8.2 Latar Belakang

Latar belakang dalam membuat aplikasi pengolahan data administrasi barang adalah dalam sistem pengambilan paket yang ada dikampus Politeknik Pos Indonesia belum tertata secara efektif dan efisien. Sehingga dibuatlah aplikasi ini yang dapat membantu memudahkan petugas satpam dalam pengimputan data paket serta dari segi mahasiswa dalam pengambilan paket lebih diproses lebih cepat dengan data yang valid. Sehingga resiko terjadinya tertukarnya barang sangat kecil.

8.3 Identifikasi Masalah

Berdasarkan latar belakang masalah yang ada, maka dapat diidentifikasi menjadi beberapa masalah sebagai berikut:

1. Bagaimana penerapan dari aplikasi pengolahan data administrasi barang mencari data barang ?

8.4 Tujuan Dan Manfaat

Berdasarkan latar belakang dan identifikasi masalah yang ada, maka akan memberikan tujuan dan manfaat sebagai berikut:

1. Meningkatkan efektitas dan efisiensi cara kerja dari segi petugas dan mahasiswa saat melakukan proses pengambilan barang.

8.5 Ruang Lingkup

Dari data-data yang didapatkan dari proses identifikasi masalah ini maka ruang lingkup dari pembahasan ini adalah sebagai berikut:

1. Kampus Politeknik Pos Indonesia.

8.6 Sistematika Penulisan

Materi-materi yang tertera pada Laporan Proyek II ini dikelompokkan menjadi beberapa sub bab dengan sistematika penyampaian sebagai berikut :

1. BAB 1 PENDAHULUAN Pada bab ini berisi latar belakang, identifikasi masalah ,tujuan ,ruang lingkup dan sistematika penulisan.
2. BAB 2 LANDASAN TEORI Pada bab ini berisi tentang teori yang berupa pengertian atau definisi yang dikutip dari jurnal.
3. BAB 3 METODE Pada bab ini akan dijelaskan fungsi algoritma dan bahasa pemrograman yang digunakan pada program / sistem aplikasi dan akan dijelaskan bagaimana alur kerja program aplikasi yang akan digambarkan dalam bentuk use case sehingga pembaca akan lebih mudah mengerti dalam hal ini.
4. BAB 4 PERCOBAAN DAN HASIL Bab ini menjelaskan tentang percobaan pembuatan aplikasi yang terdiri dari alat-alat pendukung, lingkungan implementasi, tampilan antar muka, dan petunjuk pemakaian. Bab ini juga membahas tentang pembahasan hasil pembuatan aplikasi.

5. BAB 5 KESIMPULAN DAN SARAN Bab ini berisi kesimpulan dan saran yang berkaitan dengan analisis yang telah dituliskan pada bab sebelumnya sehingga pembaca lebih mudah mengerti hasil dari analisis yang telah dilakukan dan penulis juga dapat memberikan saran di bab ini.

BAB 9

LANDASAN TEORI

9.1 Tinjauan Pustaka

Dalam suatu penelitian diperlukan dukungan hasil penelitian yang telah ada sebelumnya yang berkaitan dengan penelitian tersebut. Menurut Sutabri Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. pengolahan data adalah suatu proses menerima dan mengeluarkan data menjadi bentuk lain yaitu berupa informasi.Pengolahan Data adalah manipulasi data agar menjadi bentuk yang lebih berguna. Pengolahan data tidak hanya berupa perhitungan numerik tetapi juga operasi- operasi seperti klasifikasi data dan perpindahan data dari satu tempat ke tempat lain. Menurut Siagianadministrasi adalah keseluruhan proses kerjasama antara dua orang manusia atau lebih yang didasarkan atas rasionalitas tertentu untuk mencapai tujuan yang telah ditentukan sebelumnya[11].Administrasi sebagai usaha dan kegiatan yang berkaitan dengan penyelenggaraan kebijakan untuk mencapai tujuan.Administrasi didenisikan sebagai keseluruhan proses kerjasama antara dua orang manusia atau lebih yang didasarkan atas rasionalitas tertentu untuk mencapai tujuan yang telah ditentukan sebelumnya. Dengan demikian, dapat disimpulkan bahwa pengolahan data administrasi barang merupakan kegiatan menyimpan

data serta penanganan data guna untuk disusun dan dicatat untuk mencapai tujuan yang telah ditentukan

9.2 Oracle Apex

Oracle Application Express (Oracle APEX) yang dulu disebut HTML-DB adalah sebuah framework yang berbasis pada sebuah database dedicated (sementara ini sampai versi terbaru masih untuk Oracle DB saja dan lisensi include dalam lisensi database, ini artinya apa bahwa engine aplikasi dibangun sepenuhnya didalam sebuah database [5]. Bahkan untuk arsitektur seperti yang dipakai dalam Oracle XE dan Oracle 11G le image (library,css,theme,dll) disimpan didalam database metadata.

9.3 Konsep Dasar Aplikasi

Pengertian aplikasi adalah program yang dibuat dengan tujuan untuk melaksanakan fungsi sesuai dengan kegunaan aplikasinya, penggunanya dan jenis aplikasi itu sendiri [3]. Aplikasi dibuat dengan bahasa pemrograman yang bertujuan untuk membantu memecahkan masalah dengan aturan yang sesuai dengan bahasa pemrograman itu sendiri yang nantinya bisa mengolah data.

9.4 Sistem Informasi Berbasis Website

Pengertian website adalah sekumpulan halaman berisi informasi yang berada di WWW (World Wide Web) di dalam internet [2]. Website bisa diakses jika perangkat yang digunakan terhubung ke internet. Website mempermudah dalam pencarian informasi yang dibutuhkan untuk keperluan tertentu.

9.5 Web Mobile

Pengertian web mobile adalah website yang bisa diakses dan dilihat menggunakan perangkat seperti smartphone. Web mobile bersifat eksibel karena bisa diakses dimanapun dan kapanpun sehingga pengguna lebih mudah untuk mencari informasi yang dibutuhkan hanya dengan perangkat mobile [4].

9.6 Algoritma

Pengertian adalah langkah-langkah logis yang bertujuan untuk menyelesaikan suatu masalah dengan tepat. Algoritma sangat dibutuhkan dalam pembuatan aplikasi dan juga pengolahan data di dalam komputer [5]. Tidak hanya dalam aplikasi, algoritma juga kita terapkan dalam kehidupan sehari-hari, seperti langkah-langkah dalam

memasak, langkah-langkah dalam menyuci pakaian, dan masih banyak lagi. Algoritma yang akan diterapkan dalam pembuatan aplikasi ini adalah bagaimana cara terbentuknya aplikasi pengolahan data administrasi barang menggunakan aplikasi apex online.

9.7 Perancangan Sistem

Analisis sistem adalah penelitian terhadap sebuah sistem dengan tujuan untuk melakukan pengembangan sistem itu sendiri dan bisa menjadi sebuah pembelajaran terhadap sistem yang diteliti [1]. Perancang penting dalam segala aspek dan sangat dibutuhkan untuk melakukan sebuah penelitian maupun pengembangan. Kemampuan dalam merancang suatu sistem juga diperlukan saat mempelajari bagaimana proses terciptanya aplikasi dan cara kerja sebuah aplikasi dalam mengolah data.

9.8 Use Case

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara aktor inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana. Perilaku sistem adalah bagaimana sistem akan beraksi. Perilaku ini merupakan aktivitas sistem yang bisa dilihat dari luar dan bisa diuji. Perilaku sistem ini dicapture di dalam Use Case. Use Case sendiri mendeskripsikan sistem, lingkungan sistem, serta hubungan antara sistem dengan lingkungannya. Deskripsi dari sekumpulan aksi sekuenzial yang ditampilkan sistem yang menghasilkan yang tampak dari nilai ke actor khusus. Use Case digunakan untuk menyusun behavioral things dalam sebuah model. Pada gambar berikut contoh simbol Use Case.

9.9 Activity Diagram

Diagram aktitas atau activity diagram menggambarkan workow (aliran kerja) atau aktitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. yang perlu diperhatikan disini adalah bahwa diagram aktitas menggambarkan aktitas sistem bukan apa yang dilakukan aktor, jadi aktitas yang dapat dilakukan oleh sistem.

9.10 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan skenario untuk rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan suatu output tertentu.

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (energi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 9.1 Use Case

Simbol	Deskripsi
status awal	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / decision	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / join	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
fork,	digunakan utk menunjukkan kegiatan yg dilakukan secara parallel
join,	digunakan utk menunjukkan kegiatan yg digabungkan

Gambar 9.2 Activity Diagram3

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Gambar 9.3 Activity Diagram3

BAB 10

ANALISIS DAN PERANCANGAN

10.1 Data

Data merupakan sekumpulan informasi atau juga keterangan keterangan dari suatu hal yang diperoleh dengan melalui pengamatan atau juga pencarian ke sumber sumber tertentu. Data yang diperoleh tersebut dapat menjadi sebuah anggapan atau fakta disebabkan karena memang belum diolah dengan lebih lanjut. Setelah data diolah dengan melalui penelitian atau percobaan maka suatu data tersebut dapat menjadi bentuk yang lebih kompleks seperti sesuatu database, informasi atau juga bahkan solusi untuk menyelesaikan masalah tertentu. Pada bagian ini, akan dibahas mengenai aplikasi pengolahan data administrasi barang menggunakan aplikasi apex oracle .Dari aplikasi ini akan digambarkan dalam bentuk use diagram, activity diagram dan sequence diagram.

10.2 Sistem yang sedang berjalan

Untuk gambaran sistem yang berjalan dalam aplikasi pengolahan data administrasi barang adalah proses dimana pengambilan barang di kantor pos Politeknik Pos Indonesia yang dibuat dengan Oracle APEX.

10.3 Dokumen yang digunakan

Dokumen yang digunakan untuk menjelaskan dokumen yang digunakan dalam proses penyampaian informasi pada aplikasi. Berikut merupakan analisa dokumen yang menjelaskan hal - hal yang akan dilakukan dalam tabel seperti berikut.

Dibuat oleh	Admin Aplikasi
Dibuat untuk	User
Isi	List yang berisi informasi yang diperbarui atau ditambahkan.
Frekuensi	Dibuat berdasarkan informasi terbaru sesuai dengan rekam data barang.
Tujuan	Menyampaikan informasi data barang terbaru sesuai dengan rekam data barang.

Gambar 10.1 Activity Diagram3

10.4 Sistem yang dibangun

Pada sistem ini dimaksudkan ke dalam use case mengenai sistem yang akan berjalan pada Aplikasi Pengolahan Data Administrasi Barang.

10.5 Sistem yang Akan Dibangun pada penyampaian Informasi Terbaru

Dalam aplikasi ini customer/ user mencari informasi yang akan dibutuhkan dengan cara menunjukan identitas, input kode resi barang dan untuk mengetahui barang paket telah diambil atau belum diambil harus melalui aplikasi ini menggunakan database yang telah tersedia. Sehingga kami membuat sistem pengambilan barang yang berada di kantor satpam Politeknik Pos Indonesia yang awal mulanya masih menggunakan cara manual sekarang sudah mengalami perkembangan dengan memanfaatkan kecanggihan teknologi dengan menggunakan Apex Oracle Online.

10.6 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berhubungan dengan sistem yang berjalan. Dimana akan menjabarkan tentang fungsi-fungsi yang akan mendukung jalannya sistem. Adapun kebutuhan fungsional yang akan dibuat ada 2 yaitu berikut :

1. Fungsi Search
2. Edit
3. Create

10.7 Kebutuhan Perangkat Keras dan Perangkat Lunak

1. Berikut adalah spesifikasi Kebutuhan Perangkat Keras yang digunakan pada tabel III.2
2. Berikut adalah spesifikasi Kebutuhan Perangkat Lunak yang digunakan pada tabel III.3

10.8 Kebutuhan Non-Fungsional

Kebutuhan perangkat lunak dilakukan untuk mengetahui kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras (hardware) dan analisis perangkat lunak (software).

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	500 GB	Media untuk menyimpan data aplikasi yang dibuat
2	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3	<i>Processor</i>	<i>Intel® Pentium(R) CPU N4200 @1,10 Ghz</i>	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur Jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

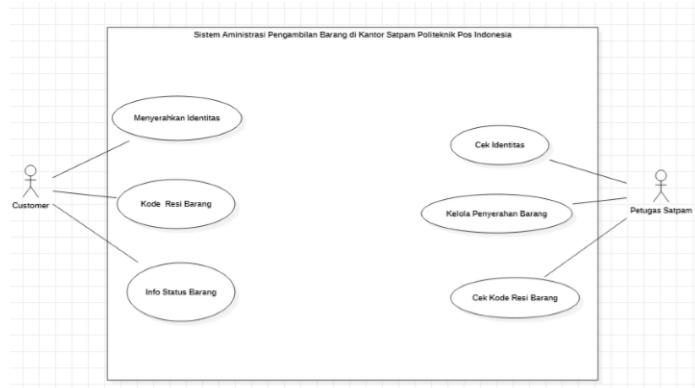
Gambar 10.2 Perangkat User

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	500 GB	Media untuk menyimpan data aplikasi yang dibuat
2	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3	<i>Processor</i>	<i>Intel® Pentium(R) CPU N4200 @1,10 Ghz</i>	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur Jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

Gambar 10.3 Perangkat Lunak User

10.9 Use Case

sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara aktor inisiatör dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana. Diagram Use Case merupakan gambaran garis dari interaksi diantara unsur - unsur sistem. Hal ini digunakan dalam menganalisis sistem untuk mengidentifikasi, menjelaskan dan mengatur persyaratan sistem. Diagram Use Case menggambarkan proses apa saja yang dilakukan oleh aktor. Aktor untuk level tertentu memiliki batasan untuk mengakses aplikasi ini. Pada Diagram use case pada gambar dibawah ini ini:

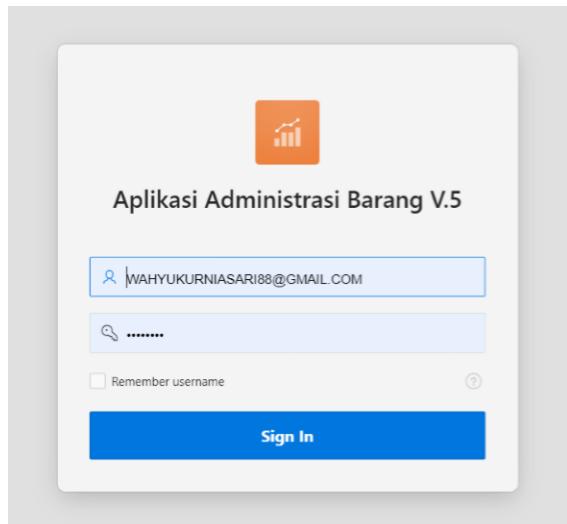
**Gambar 10.4** Use Case Sistem Administrasi Pengambilan Barang

perangkat informasi sehingga pengguna dapat lebih mudah melakukan interaksi dengan program. Dalam merancang sebuah user interface kita harus memperhatikan beberapa hal :

1. User Interface yang sederhana, maksudnya di dalam aplikasi tersebut tidak ada elemen fungsi yang kiranya tidak diperlukan dan hanya terdapat elemen yang dibutuhkan oleh pengguna
2. User Interface juga dibentuk sesuai siapa pengguna aplikasi tersebut. Sehingga dapat menyesuaikan kemampuan dan pengalaman user dalam berinteraksi.
3. Penempatan layout agar tidak mempersulit pengguna dalam menggunakan sebuah aplikasi

10.10 Perancangan Antar Muka yang Dibangun

Dalam perancangan antar muka yang akan dibangun pada aplikasi Pengolahan Data Administrasi Barang Menggunakan Apex Online Aplikasi ini sudah hampir tidak ada yang perlu dibangun lagi, karena sudah memenuhi hal yang perlu diperhatikan dalam merancang User Interface, hanya saja disini kita memperbaikannya menyesuaikan seperti user interface yang ada pada aplikasi pada umumnya di zaman sekarang ini. Contohnya seperti user interface yang ada pada aplikasi pada aplikasi dibawah ini merupakan use case dari sistem administrasi pengambilan barang di kantor satpam Politeknik Pos Indonesia yaitu pada sistem customer melakukan beberapa tahap seperti menyerahkan identitas, menunjukkan kode resi barang, serta dapat mengetahui info status barang sedangkan, pada sistem petugas satpam itu sendiri adalah melakukan cek identitas dalam tahap ini petugas akan mengecek validasi data terhadap identitas dan barang yang akan diambil oleh customer sehingga meminimalisir terjadi kekeliruan barang, Selain itu petugas tugas mengelola penyerahan barang apakah barang yang sudah ada telah diambil oleh customer ataupun belum, yang terakhir adalah cek kode resi barang, pada tahapan ini petugas akan melakukan pengecekan kode resi pada barang yang telah ada.



Gambar 10.5 Tampilan User Interface Login yang Telah dibuat



Gambar 10.6 Tampilan User Interface Login yang Telah dibuat

BAB 11

IMPLEMENTASI DAN PERANCANGAN

11.1 Lingkungan Implementasi

Sesudah menyelesaikan proses perancangan , proses yang dilakukan selanjutnya adalah percobaan. Tahap ini merupakan tahapan peletakan sistem agar dapat dioperasikan sehingga perlu penyesuaian terhadap rancangan yang meliputi lingkungan perangkat keras dan perangkat lunak

11.2 Lingkungan Perangkat keras

Dalam pembuatan aplikasi ini, spesifikasi perangkat keras yang digunakan adalah sebagai berikut

1. Processor : Intel R CoreTM i5-8300H up to 4.00Ghz
2. Memory : 8GB DDR4
3. Harddisk : 1TB HDD

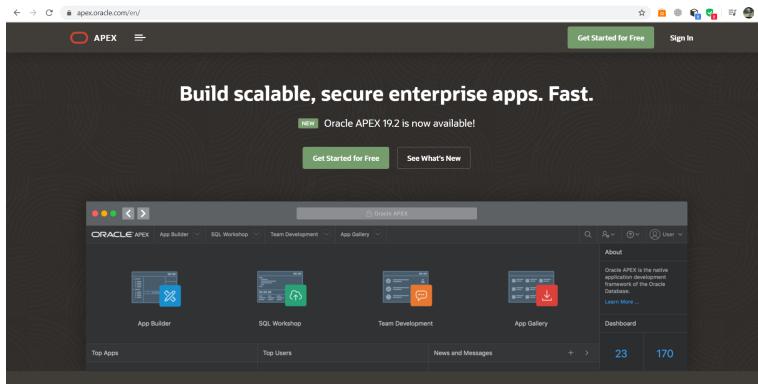
Dalam menganalisis aplikasi ini, spesifikasi perangkat lunak yang digunakan adalah sebagai berikut

1. Sistem Operasi : Windows 10
2. Bahasa Pemrograman : Java
3. Software Pendukung : Oracle Application Express (APEX)

11.2.1 Langkah Kerja

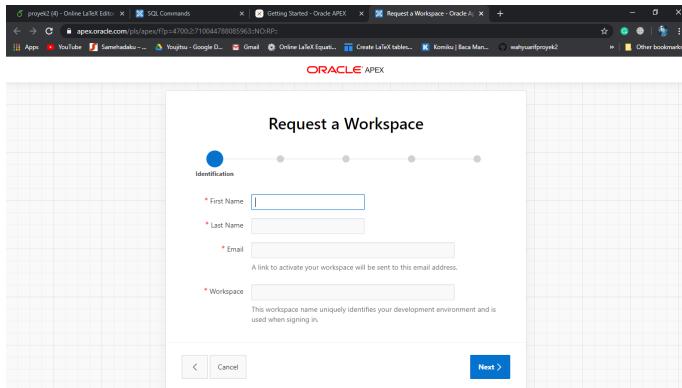
Dalam pembuatan aplikasi oracle express, kita perlu mengakses situs web *apex.oracle.com* dan selanjutnya membuat dan mendapatkan workspace secara gratis dari *oracle*. Pastikan terhubung dengan email yang aktif untuk melakukan registrasi untuk melakukan pembuatan aplikasi ini. Berikut merupakan tahapan pembuatan aplikasi pengelolaan data administrasi barang menggunakan oracle apex online

1. Pertama adalah kita harus melakukan registrasi pada website *oracle express* yaitu <https://apex.oracle.com> lalu klik *Get Start For Free* pada pojok kanan atas untuk mendapatkan *free workspace*.



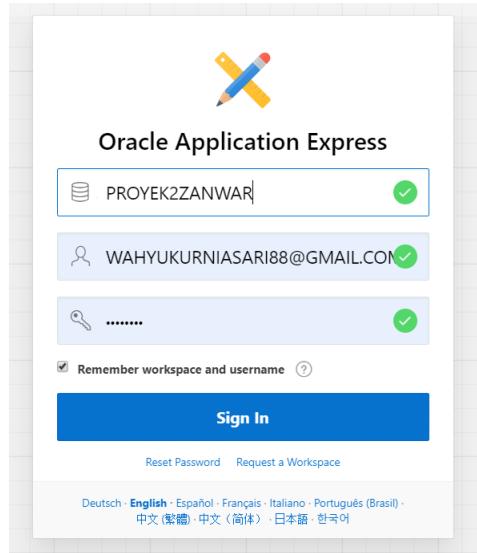
Gambar 11.1 Halaman Utama Pada apex oracle.com

2. Kemudian klik *Request a Free Workspace* dan kita akan mengisi form yang untuk mendapatkan workspace kita seperti pada gambar berikut.



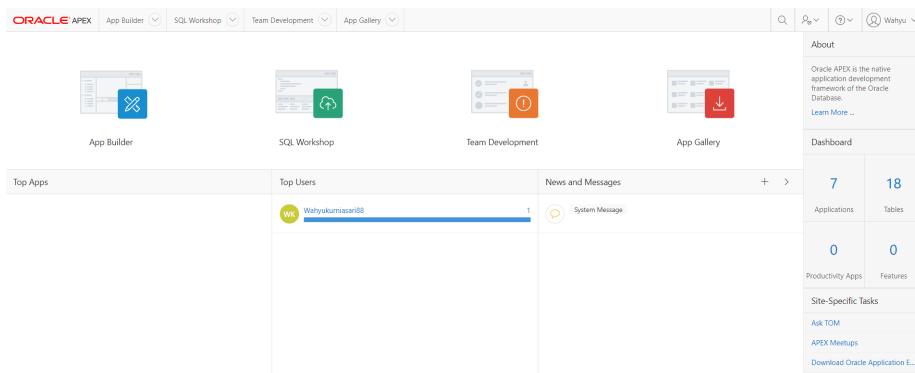
Gambar 11.2 Form Request a Workspace

3. Selanjutnya yaitu melakukan *Sign In Workspace* atau lembar kerja yang sudah dibuat, dengan mengisi nama workspace, email serta password. Apabila sudah mengisi *workspace*, *email*, dan *password* klik *sign in* untuk melakukan tahapan selanjutnya.



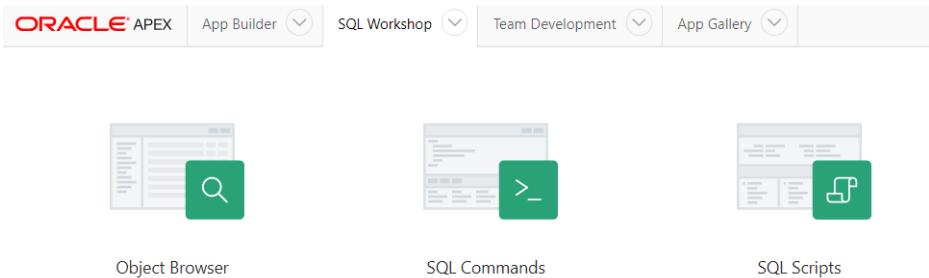
Gambar 11.3 Halaman Sign In Apex Oracle

4. Tahapan selanjutnya adalah apabila sudah melakukan *sign in* maka kita akan masuk pada halaman utama workspace seperti pada gambar dibawah. Untuk membuat aplikasi tentunya kita harus membuat tabel terlebih dahulu, caranya kita klik pada *Sql Workshop* untuk melihat menu lebih lanjut.



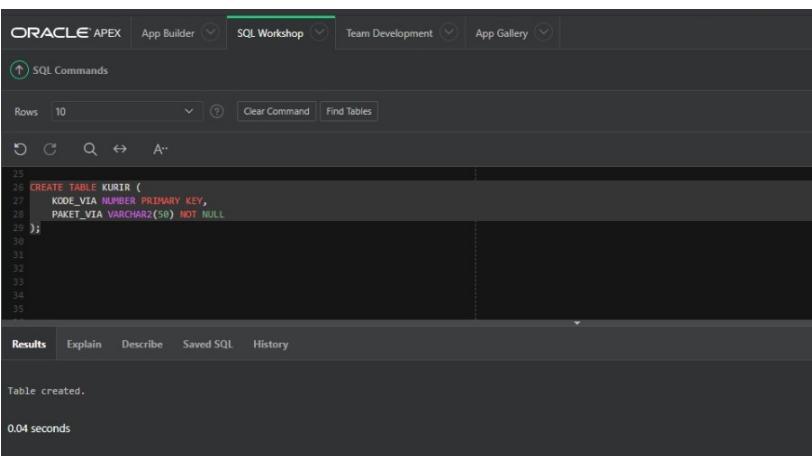
Gambar 11.4 Menu Pada Apex Oracle

5. Setelah melakukan klik pada *Sql Workshop* akan muncul beberapa menu seperti *object browser*, *sql command*, *sql scripts*. Pada pembuatan aplikasi pengolahan data administrasi barang ini kita menggunakan *Sql Command* untuk menuliskan perintah pemembuatan tabel yang akan kita gunakan untuk aplikasi yang akan kita buat.



Gambar 11.5 Menu Pada SQL Workshop

6. Tahapan selanjutnya yaitu perlu dilakukan untuk membuat sebuah aplikasi pengelolaan data administrasi barang adalah menormalisasi tabel-tabel yang akan dibuat. Tujuan dilakukannya normalisasi tabel adalah untuk menghilangkan dan mengurangi redundansi data dan tujuan yang kedua adalah memastikan dependensi data (Data berada pada tabel yang tepat). Setelah data sudah berada pada tabel yang tepat maka langkah selanjutnya adalah *create tabel* seperti gambar 11.6 pembuatan tabel kurir. Dalam pembuatan tabel kurir ini disertai dengan pemilihan tipe data seperti tipe data *VARCHAR2* adalah tipe data untuk menampung data yang bertipe karakter atau *alfabetic(a-z)* seperti pada gambar 11.6 ini misalnya pada atribut paket_via menggunakan tipe data *VARCHAR2*. Selain itu pada tipe data *NUMBER* digunakan untuk menyimpan data dalam bentuk angka dimana pada atribut kode_kurir kita gunakan sebagai *primary key*.



The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. In the SQL Commands editor, the following SQL code is displayed:

```
25
26 CREATE TABLE KURIR (
27   KODE_VIA NUMBER PRIMARY KEY,
28   PAKET_VIA VARCHAR2(50) NOT NULL
29 );
```

The results pane below shows the output:

```
Table created.
```

Execution time: 0.04 seconds

Gambar 11.6 Pembuatan Tabel Paket

7. Tahapan selanjutnya adalah membuat tabel pegawai. Pada tabel pegawai ini harus menginputkan data-data yang diperlukan seperti atribut kode pegawai, nama pegawai, alamat, dan no hp. Pada proses pembuatan tabel ini kolom kode pegawai menjadi *primary key*. Tipe data *primary key* merupakan suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam tabel. *primary key* juga sebuah aturan dimana fungsinya adalah untuk membedakan antara baris satu dengan baris lainnya yang ada pada tabel dan bersifat unik. Apabila tabel pegawai sudah dibuat maka tinggal dijalankan. Dan jika sql command kita sudah benar maka tabel pegawai dirun/dijalankan maka akan muncul tulisan *table created* seperti pada gambar 11.7 dibawah.

The screenshot shows the Oracle APEX interface with the 'SQL Workshop' tab selected. In the SQL Commands pane, the following SQL code is displayed:

```
21
22 CREATE TABLE PEGAWAI (
23     KODE_PEGAWAI VARCHAR2(50) PRIMARY KEY,
24     NAMA_PEGAWAI VARCHAR(50),
25     ALAMAT VARCHAR2(50),
26     NO_HP NUMBER
27 );
```

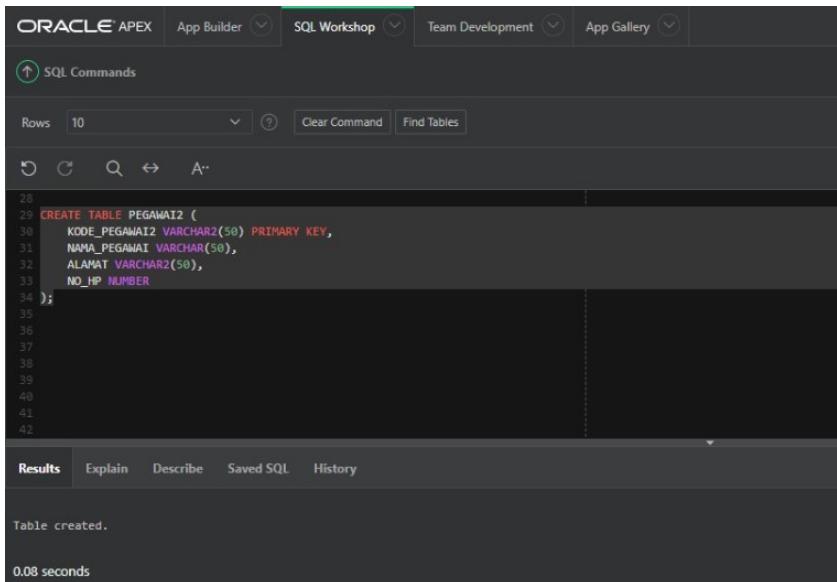
The Results pane below shows the output:

```
Table created.
```

Execution time: 0.16 seconds

Gambar 11.7 Pembuatan Tabel Pegawai

8. Proses berikutnya adalah masih dalam tahapan pembuatan tabel. Pada langkah ketiga ini adalah pembuatan tabel pegawai lagi. Karena pada tabel utama yang akan kita gunakan membutuhkan 2 data pegawai oleh karena itu kita membuatnya dengan atribut data yang sama seperti pada tabel pegawai sebelumnya, hanya saja kita membedakan nama tabelnya menjadi pegawai2, dan atribut kode_pegawai yang kita buat menjadi kode_pegawai2 dan kita jadikan primary key juga seperti pada gambar 11.8.



The screenshot shows the Oracle APEX interface with the 'SQL Workshop' tab selected. In the main area, a SQL command is being entered to create a table:

```
28
29 CREATE TABLE PEGAWAI2 (
30   KODE_PEGAWAI2 VARCHAR2(50) PRIMARY KEY,
31   NAMA_PEGAWAI VARCHAR(50),
32   ALAMAT VARCHAR2(50),
33   NO_HP NUMBER
34 );
```

The command is numbered from 28 to 42. Below the code, the results of the execution are shown:

Table created.
0.08 seconds

Gambar 11.8 Pembuatan Tabel Pegawai2

9. Tahapan selanjutnya adalah pembuatan tabel barang. Berikut ini adalah pembuatan tabel barang dimana atribut kode_resi sebagai primary key. Hanya saja pada tabel ini terdapat atribut yang sama seperti atribut yang ada pada tabel sebelumnya, yaitu kode_via, kode_pegawai, dan kode_pegawai2 yang akan dijadikan *foreign key*. Dimana *foreign key* adalah *primary key* ke-2 yang berfungsi untuk merelasikan tabel barang dengan tabel lain yang memiliki *parent key* dari atribut *foreign key* ditabel barang tersebut. Pada gambar 11.9 berikut adalah sql command untuk membuat tabel barang dan *constraint foreign key*-nya.

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The code area contains the SQL command to create the BARANG table, including primary key and foreign key constraints:

```
1 CREATE TABLE BARANG (
2     KODE_RESI VARCHAR2(50) PRIMARY KEY,
3     TGL_BLN_TH DATE NOT NULL,
4     KODE_VIA NUMBER NOT NULL,
5     NAMA_PENGIRIM_PAKET VARCHAR2(50) NOT NULL,
6     NAMA_PENERIMA_PAKET VARCHAR2(50) NOT NULL,
7     KODE_PEGAWAI VARCHAR2(50) NOT NULL,
8     STATUS VARCHAR2(50) NOT NULL,
9     DIAMBIL_OLEH VARCHAR2(50),
10    KODE_PEGAWAI2 VARCHAR2(50),
11    TANGGAL_AMBIL DATE,
12    CONSTRAINT FK_VIA1 FOREIGN KEY (KODE_VIA) REFERENCES KURIR,
13    CONSTRAINT FK_PEGAWAI1 FOREIGN KEY (KODE_PEGAWAI) REFERENCES PEGAWAI,
14    CONSTRAINT FK_PEGAWAI2 FOREIGN KEY (KODE_PEGAWAI2) REFERENCES PEGAWAI2
15 );
16
17
18
```

The results section shows the message "Table created." and a execution time of "0.14 seconds".

Gambar 11.9 Data Pada Tabel Barang

10. Sebelumnya lanjut ke proses pembuatan aplikasi. Pertama dikarenakan pada tabel Kurir kita menggunakan tipe data *number* pada atribut kode_kurir maka tidak pada menu form pengisian atribut apex tidak menampilkan field untuk menginputkan data number, maka kita perlu membuat fungsi *Auto Increment* dengan membuat sequence yang kemudian kita panggil kedalam fungsi trigger seperti pada gambar 11.10.

```

1 CREATE SEQUENCE INC_KODE MINVALUE 1 MAXVALUE 999 INCREMENT BY 1;
2
3 CREATE OR REPLACE TRIGGER AUTO_INC1
4 BEFORE INSERT ON KURIR
5 FOR EACH ROW
6 BEGIN
7   SELECT INC_KODE.NEXTVAL
8   INTO  :NEW.KODE_VIA
9   FROM  dual;
10 END;
11

```

Gambar 11.10 Trigger Auto Increment

11. Kedua dikarenakan disini kita memiliki 2 tabel pegawai dan akan hanya ada 1 tabel pegawai yang akan ditampilkan di aplikasi, maka kita perlu membuat fungsi *trigger* terlebih dahulu yang dimana berfungsi apabila terjadi *create*, *update*, dan *delete* pada tabel pegawai, maka akan terjadi hal yang serupa pada tabel pegawai2. Pada gambar 11.11 berikut adalah sql command pembuatan trigger pada tabel pegawai.

```

1 CREATE OR REPLACE TRIGGER TRIGGER_PEGAWAI
2 AFTER INSERT OR UPDATE OR DELETE ON PEGAWAI
3 FOR EACH ROW
4 BEGIN
5   IF INSERTING THEN
6     INSERT INTO PEGAWAI2 VALUES (:NEW.KODE_PEGAWAI, :NEW.NAMA_PEGAWAI, :NEW.ALAMAT, :NEW.NO_HP);
7   END IF;
8
9   IF UPDATING THEN
10    UPDATE PEGAWAI2
11    SET KODE_PEGAWAI2 = :NEW.KODE_PEGAWAI,
12        NAMA_PEGAWAI = :NEW.NAMA_PEGAWAI,
13        ALAMAT = :NEW.ALAMAT,
14        NO_HP = :NEW.NO_HP;
15    END IF;
16
17   IF DELETING THEN
18    DELETE FROM PEGAWAI2 WHERE KODE_PEGAWAI2 = :OLD.KODE_PEGAWAI;
19   END IF;
20 END;

```

Gambar 11.11 Create Trigger Pegawai

12. Di aplikasi ini kita juga menampilkan *view* dimana data yang ditampilkan *view* akan bisa didownload oleh user, dan berguna sebagai rekap data *hardcopy*. Pada *view* ini hanya akan menampilkan beberapa atribut yang ada pada setiap tabel, pada gambar 11.12 berikut adalah sql command untuk membuat *view*.

The screenshot shows the Oracle APEX interface with the "SQL Workshop" tab selected. In the main area, there is a code editor containing the following SQL code:

```

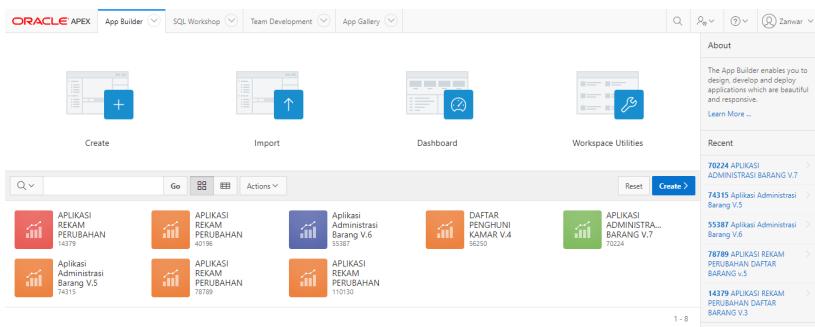
1 CREATE OR REPLACE FORCE EDITIONABLE VIEW "LAPORAN_PAKET" AS
2 SELECT kode_resi, TGL_BLM_THN AS tanggal_masuk, pegawai.kode_pegawai,
3 pegawai.nama_pegawai AS pegawai_yang_menerima, tanggal_ambil AS tanggal_keluar, pegawai2.kode_pegawai2,
4 diambil_oleh, pegawai2.nama_pegawai AS pegawai_yang_memberi
5 FROM BARANG
6 INNER JOIN pegawai ON barang.kode_pegawai = pegawai.kode_pegawai
7 INNER JOIN pegawai2 ON barang.kode_pegawai2 = pegawai2.kode_pegawai2

```

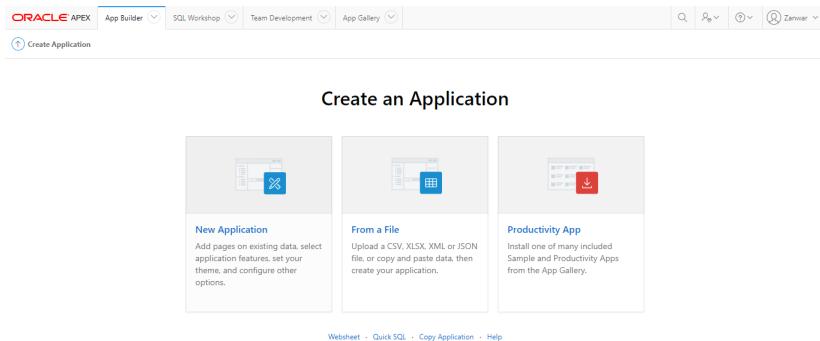
Below the code editor, the "Results" tab is selected, showing the message "View created." and a execution time of "0.25 seconds".

Gambar 11.12 Create View

13. Setelah menyiapkan tabelnya, sekarang kita bisa masuk ke proses pembuatan aplikasi, dengan cara klik "App Builder" disini kita akan menemukan beberapa menu seperti pada gambar 11.13, yaitu *Create*, *Import*, *Dashboard*, dan *Workspace utilities*. Karena disini kita akan membuat aplikasi maka pilih menu *Create*.

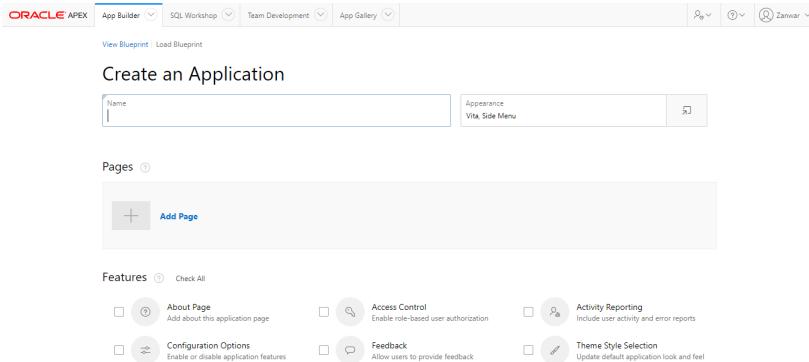
**Gambar 11.13** Menu App Builder

14. Disini kita akan ada 3 menu, yaitu *New Application* digunakan untuk membuat langsung aplikasi dengan tabel yang sudah ada, *From a File* digunakan untuk membuat aplikasi dengan mengunggah file dalam bentuk CSV, XLSX, XML, atau JSON sebagai data untuk membuat aplikasi, dan *Productivity App* digunakan untuk install salah 1 aplikasi yang ada di galeri aplikasi seperti yang ada pada gambar 11.14. Karena kita akan membuat aplikasi dan sudah memiliki tabel data maka pilih *New Application*.



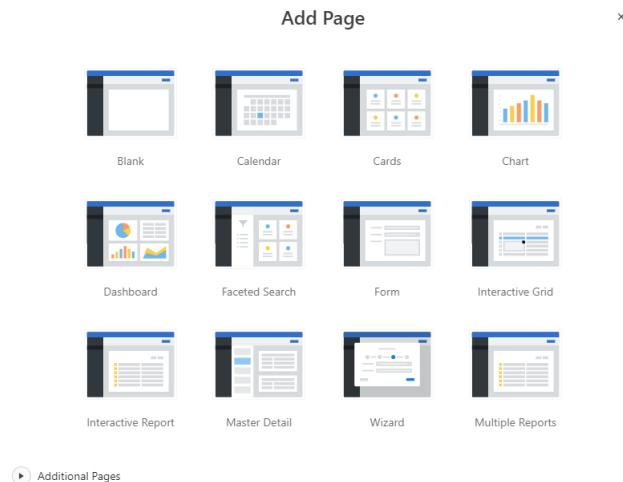
Gambar 11.14 *Menu Create Application*

15. Berikut adalah tampilan pembuatan aplikasi pada gambar 11.15.



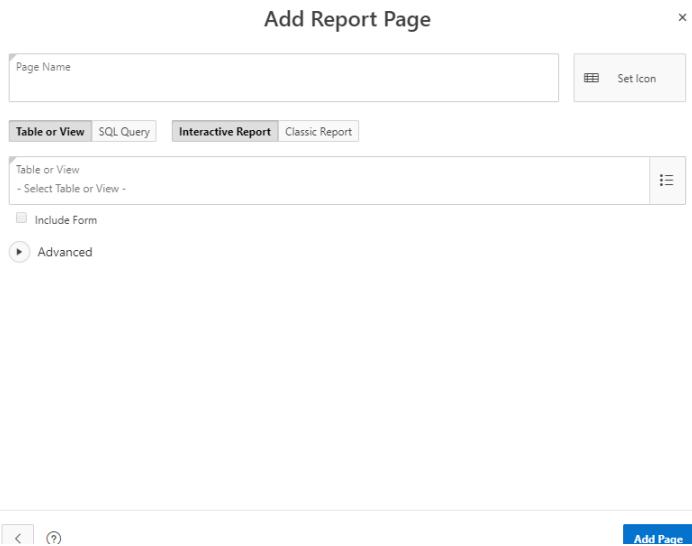
Gambar 11.15 Tampilan membuat halaman aplikasi

16. Dengan memilih *Add Page* kita diberi banyak pilihan halaman seperti apa yang ingin kita gunakan seperti pada gambar 11.16. Namun disini kita menggunakan interactive report.

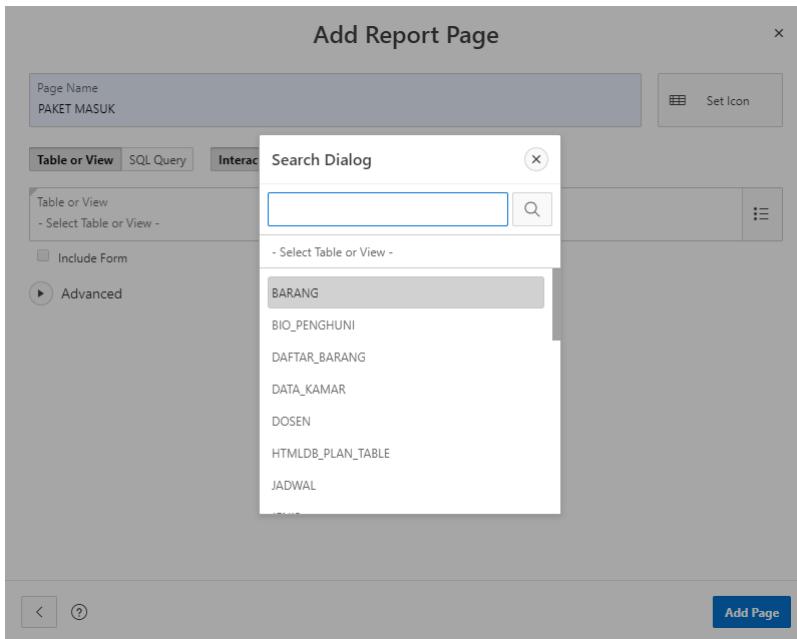


Gambar 11.16 *Add Page*

17. Pada halaman pertama kita menggunakan interactive report, berikut adalah tampilannya pada gambar 11.17. Disini kita akan membuat halaman dengan nama Paket Masuk, kemudian memilih tabel yang akan ditampilkan dengan klik di sebelah kanan tulisan -Select Table or View-. Pada gambar 11.18 adalah tampilan saat memilih tabel, kemudian pilih tabel barang.

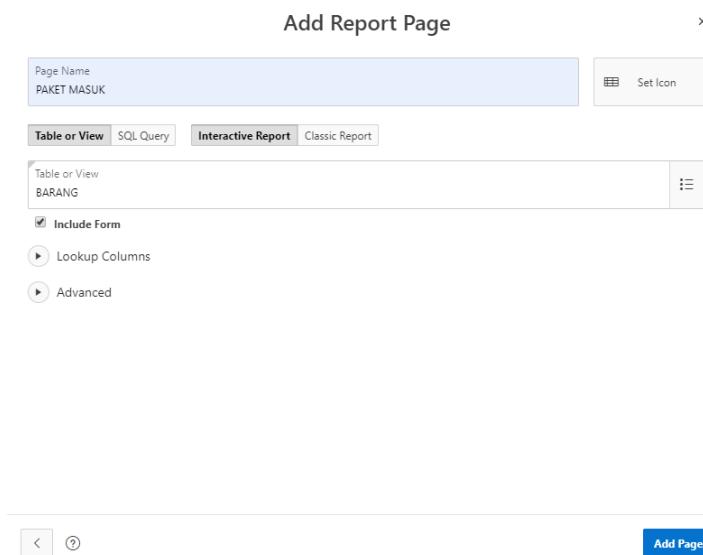


Gambar 11.17 *Interactive Report*



Gambar 11.18 *Select Table or View*

18. Berikut tampilan setelah memberi nama halaman dan memilih tabel. Kemudian untuk menyediakan form pengisian data kita dapat mengecek *Include Form* yang ada di bawah nama tabel yang sudah dipilih seperti pada gambar 11.19 berikut.



Gambar 11.19 Add Include Form

- Untuk halaman berikutnya bisa diisi dengan cara yang sama, hanya saja menampilkan tabel dan nama halaman yang berbeda, seperti pada gambar 11.20 berikut.

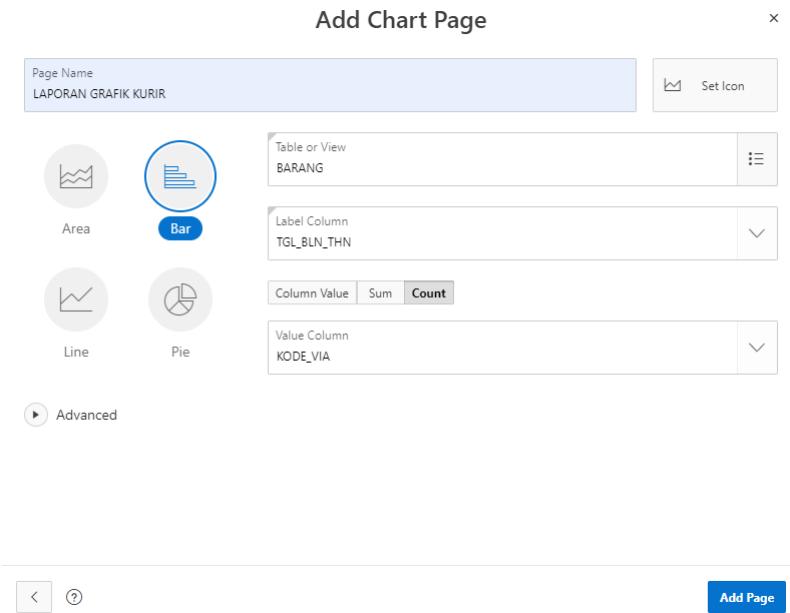
Create an Application

Name	Appearance
Home	Vita, Side Menu
PAKET MASUK	Interactive Report with Form (barang)
PAKET KELUAR	Interactive Report with Form (barang)
DAFTAR KURIR	Interactive Report with Form (kurir)
PEGAWAI	Interactive Report with Form (pegawai)

The screenshot shows the 'Create an Application' interface. It displays a list of pages with their names and appearance settings. The 'Pages' section lists five items: 'Home', 'PAKET MASUK', 'PAKET KELUAR', 'DAFTAR KURIR', and 'PEGAWAI'. Each item has an icon, a name, a type (e.g., 'Interactive Report with Form'), and edit/delete buttons.

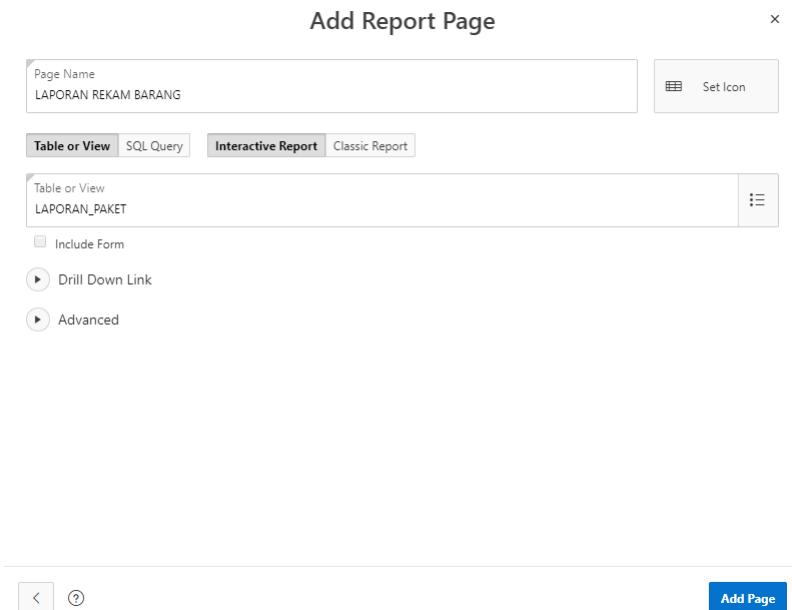
Gambar 11.20 Nama Setiap Halaman

20. Untuk halaman selanjutnya kita akan menambahkan Dashboard grafik pada tabel barang berdasarkan value dari atribut kode_via, berikut tampilan pembuatan halaman dashboard pada gambar 11.21.



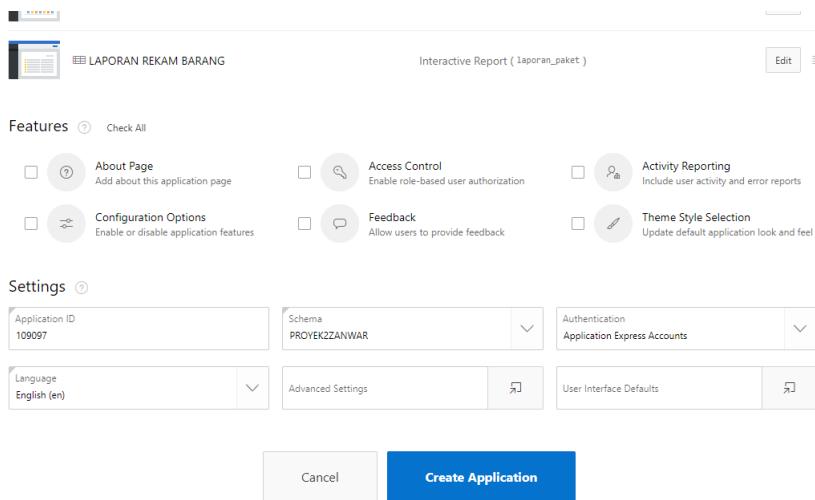
Gambar 11.21 Add Page Dashboard

21. Pada halaman berikutnya kita akan menampilkan view, caranya sama seperti menampilkan tabel sebelumnya, hanya saja disini kita memilih nama view yang sudah dibuat. Untuk tampilan view disini kita tidak bisa menambahkan form, seperti pada gambar 11.22 berikut.



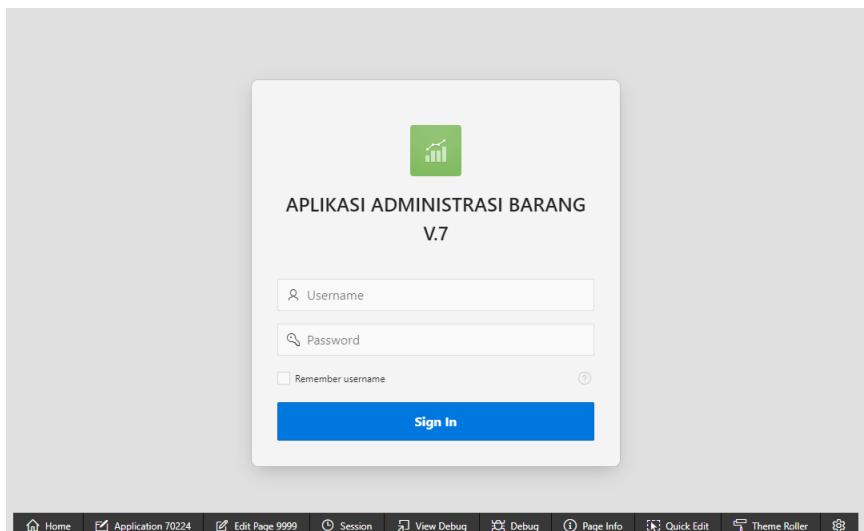
Gambar 11.22 Add Page View

22. Jika sudah membuat semua halaman kita bisa create aplikasinya dengan klik tombol *Create Application* di paling bawah halaman, seperti pada gambar 11.23 berikut.



Gambar 11.23 *Create Application* Button

23. Pada gambar 11.24 adalah tampilan awal aplikasi saat dijalankan.



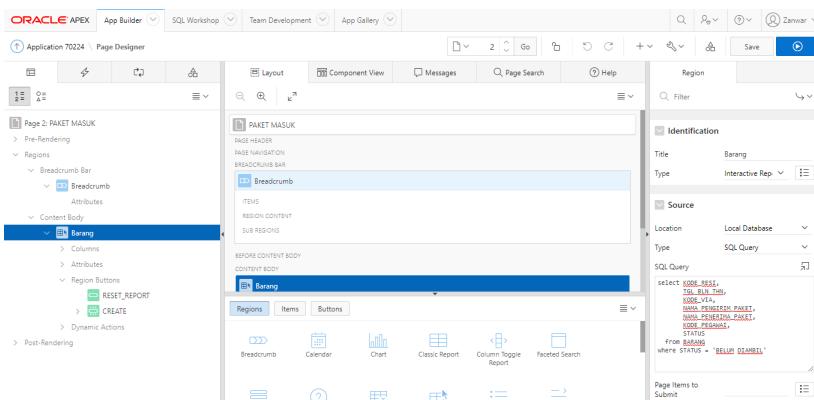
Gambar 11.24 Tampilan Login Aplikasi Administrasi Barang

24. Pada gambar 11.25 adalah tampilan halaman Paket Masuk, disini masih ada atribut yang tidak diperlukan yang tampil, kita dapat melakukan edit dengan klik tombol edit page pada menu hitam di bawah.

PAKET MASUK										Actions		
	Kode Resi / Tgl Bln Thn	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penenerima Paket	Kode Pegawai	Status	Ditambil Oleh	Kode Pegawai2	Tanggal Ambil	Reset	Create
444-121-111-567	02-NOV-2019	J&T	Permati	Sari	Aagus	SUDAH DIAMBIL	Sari	SAT1		1/19/2020		
666-091-370-491	02-NOV-2019	KANTOR POS	Sri Windari Widiastuti	Zanwar Arif Wikacsono	Cherul	SUDAH DIAMBIL	Zanwar Arif Wikacsono	SAT2		11/5/2019		
666-091-370-493	06-NOV-2019	KANTOR POS	Ratna Dewi	Ida Fatimini Putri	Aagus	BELUM DIAMBIL	-	-		-		
777-080-361-345	01-NOV-2019	JNE	Nabille	Anisa Rosalina	Aagus	SUDAH DIAMBIL	Anisa Rosalina	SAT2		11/3/2019		
777-080-361-346	03-NOV-2019	JNE	Novi Herlina Pratama	Hanif Wavian Pratama	Aagus	SUDAH DIAMBIL	Hanif Wavian Pratama	SAT2		11/7/2019		
888-079-332-250	04-NOV-2019	J&T	Lia Kumiewati	Anisa Rosalina	Budi	SUDAH DIAMBIL	Anisa Rosalina	SAT4		11/6/2019		
888-079-332-251	05-NOV-2019	J&T	Dewi Prastica Putri	Zanwar Arif Wikacsono	Cherul	SUDAH DIAMBIL	Zanwar Arif Wikacsono	SAT1		11/7/2019		
888-079-352-759	07-NOV-2019	J&T	Kiki Ristanto	Ravi Rahmatul Fajri	Budi	BELUM DIAMBIL	-	-		-		

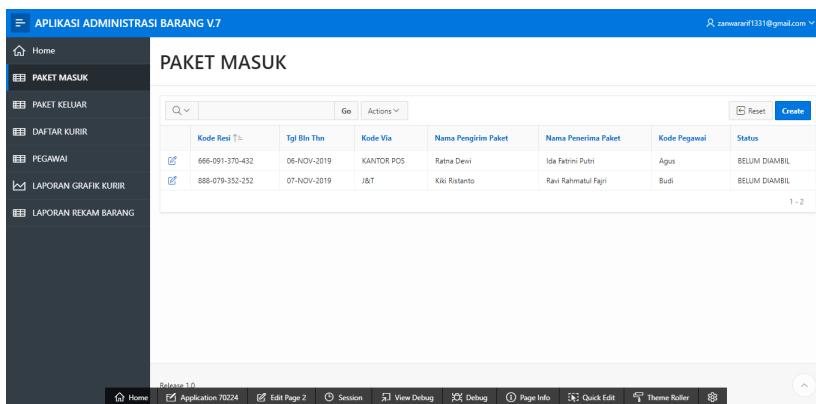
Gambar 11.25 Tampilan Paket Masuk pada Aplikasi

25. Berikut adalah tampilan edit page. Pertama pilih nama tabel Barang yang ada di *content body*, kemudian ubah *source type*-nya menjadi *SQL Query* lalu masukan query seperti pada gambar 11.26 berikut.



Gambar 11.26 Edit Page Paket Masuk

26. Setelah melakukan edit coba simpan dan jalankan aplikasinya. Berikut adalah gambar 11.27 tampilan setelah diedit.



Gambar 11.27 Paket Masuk Setelah diedit

27. Pada gambar 11.28 berikut adalah tampilan Paket Keluar, pada halaman ini juga masih ada data yang tidak perlu ditampilkan, jadi kita bisa lakukan *edit page* seperti pada halaman sebelumnya.

Kode Resi	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status	Diambil Oleh	Kode Pegawai2	Tanggal Ambil
444-121-111-567	17-JAN-2020	J&T	Permatasari	Seri Agus	SUDAH DIAMBIL	Seri Agus			19-JAN-2020
666-091-370-431	03-NOV-2019	KANTOR POS	Sri Windari Widastuti	Zainur Arif Wicaksana	Chairul	SUDAH DIAMBIL	Zainur Arif Wicaksana	Budi	05-NOV-2019
666-091-370-432	06-NOV-2019	KANTOR POS	Ratna Dewi	Ide Ferini Putri	Agus	BELUM DIAMBIL	-	-	-
777-080-361-345	01-NOV-2019	JNE	Nabilla	Anisa Rosalina	Agus	SUDAH DIAMBIL	Anisa Rosalina	Budi	03-NOV-2019
777-080-361-546	03-NOV-2019	JNE	Novi Herlina	Hanif Wavian Pratama	Agus	SUDAH DIAMBIL	Hanif Wavian Pratama	Budi	07-NOV-2019
888-079-332-250	04-NOV-2019	J&T	Lia Kurniawati	Anisa Rosalina	Budi	SUDAH DIAMBIL	Anisa Rosalina	Dono	06-NOV-2019
888-079-332-251	05-NOV-2019	J&T	Dewi Prestica Putri	Zainur Arif Wicaksana	Chairul	SUDAH DIAMBIL	Zainur Arif Wicaksana	Agus	07-NOV-2019
888-079-332-252	07-NOV-2019	J&T	Kiki Ristanto	Ravi Rahmatul Fajri	Budi	BELUM DIAMBIL	-	-	-
888-079-332-253	07-NOV-2019	J&T	Ravi Rahmatul Fajri	Budi	BELUM DIAMBIL	-	-	-	-

Gambar 11.28 Tampilan Paket Keluar pada Aplikasi

28. Lakukan hal yang sama seperti saat edit paket masuk, hanya saja masukan sql query seperti pada gambar 11.29 berikut.

```

SQL Query
SELECT KODE_BLA, TGL_BLA, KODE_VIA, NAMA_PENGIRIM_PAKET, NAMA_PENERIMA_PAKET, KODE_PEGAWAI, KODE_PEGAWAI2, STATUS, DIAMBIL_OLEH, KODE_PEGAWAI2, TANGGAL_AMBIL
FROM BASKETS
WHERE STATUS = 'SUDAH DIAMBIL'
    
```

Gambar 11.29 Edit Page Paket Keluar

29. Gambar 11.30 berikut adalah tampilan halaman Paket Keluar setelah diedit.

PAKET KELUAR										
	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status	Ditambil Oleh	Kode Pegawai2	Tanggal Ambil	
<input checked="" type="checkbox"/>	444-121-111-567	17-JAN-2020	J&T	Permatasari	Agus	SUDAH DIAMBIL	Sari	Agus	19-JAN-2020	
<input checked="" type="checkbox"/>	666-091-370-491	02-NOV-2019	KANTOR POS	Sri Windari Widiastuti	Zenwar Arif Wikaksana	CHAIRUL	SUDAH DIAMBIL	Zenwar Arif Wikaksana	Budi	05-NOV-2019
<input checked="" type="checkbox"/>	777-080-361-345	01-NOV-2019	JNE	Nabilla	Anisa Rosalina	Agus	SUDAH DIAMBIL	Anisa Rosalina	Budi	03-NOV-2019
<input checked="" type="checkbox"/>	777-080-361-346	03-NOV-2019	JNE	Novi Herlina Pratama	Agus	SUDAH DIAMBIL	Hanif Wavian Pratama	Budi	07-NOV-2019	
<input checked="" type="checkbox"/>	888-079-332-250	04-NOV-2019	J&T	Lia Kurniawati	Anisa Rosalina	Budi	SUDAH DIAMBIL	Anisa Rosalina	Dono	06-NOV-2019
<input checked="" type="checkbox"/>	888-079-332-251	05-NOV-2019	J&T	Dewi Prastica Putri	Zenwar Arif Wikaksana	CHAIRUL	SUDAH DIAMBIL	Zenwar Arif Wikaksana	Agus	07-NOV-2019
<input checked="" type="checkbox"/>	999-111-232-454	23-JAN-2020	GRAB	Anisya	Ricky	Agus	SUDAH DIAMBIL	Ricky	Agus	26-JAN-2020

Gambar 11.30 Paket Keluar Setelah diedit

30. Pada gambar 11.31 berikut adalah tampilan form pengisian data pada Paket Masuk, hanya saja pada kolom Status masih dalam bentuk *Text Field*. Untuk mengurangi resiko data yang redudansi kita perlu merubahnya menjadi *Combo Box*, seperti sebelumnya dengan klik tombol *edit page*.

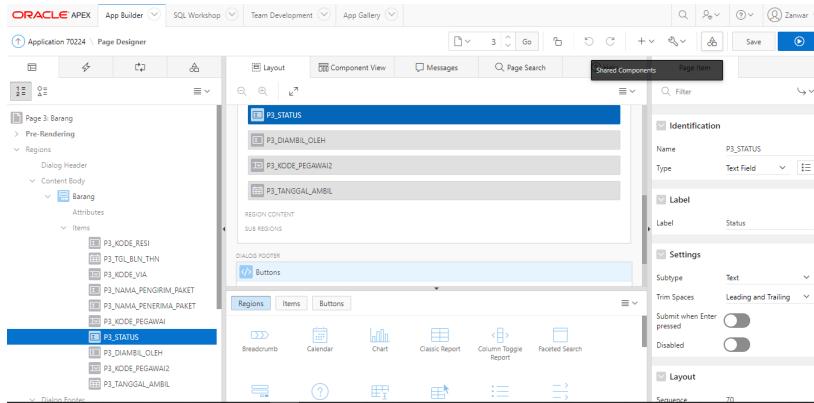
The screenshot displays a software interface for managing packages. On the left, there's a sidebar with 'MINISTRASI BARANG' at the top, followed by 'PAKE' and 'BARANG'. Below these are 'KURIR' and 'RELEASE 1.0'. The main area features a modal window titled 'Barang' with the following fields:

- Kode Resi: Tgl Bin Thn.
- Kode Via
- Nama Pengirim Paket
- Nama Penerima Paket
- Kode Pegawai
- Status
- Diambil Oleh

To the right of the modal, a sidebar titled 'Kode Pegawai' lists 'Agus' and 'Budi'. At the bottom of the screen is a navigation bar with various icons and labels: Home, Application 70224, Edit Page 3, Session, View Debug, Debug, Page Info, Quick Edit, Theme Roller, and others.

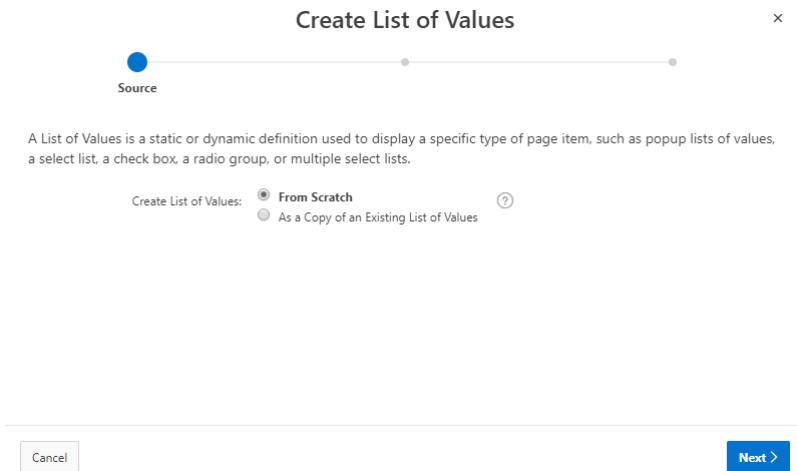
Gambar 11.31 Form Pengisian Data Paket Masuk

31. Pertama pilih tombol *Shared Components* yang ada di pojok kanan atas seperti gambar 11.32 berikut.



Gambar 11.32 Edit Page Form Paket Masuk

32. Kemudian pilih *Value of List* lalu *Create* maka kita akan masuk pada form pembuatan *Value of List* seperti gambar 11.33 berikut. Klik next lalu masukan nama dengan memilih static. Setelah itu kita dapat memasukan isi dari pilihan *Combo box* yang kita ingin seperti pada gambar 11.34 berikut.



Gambar 11.33 Create Value of List

The screenshot shows the second step of the wizard titled 'Create List of Values'. The progress bar now has four dots, all green, indicating completion. The fourth dot is blue and labeled 'Static Values'. The main area contains the following text:
Enter static display and return values. Values will display in the order entered. **Return Value** does not display, but is the value that is returned to the Application Express engine. If you do not specify a **Return Value** then it is equal to the **Display Value**. You can display additional attributes including build option controls and item level conditional display by editing the List of Values.

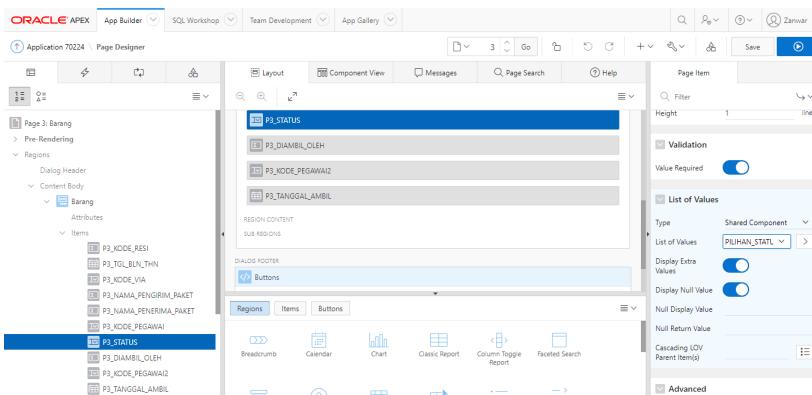
Below this is a table titled 'List of Values Name: PILIHAN_STATUS (?)'. The table has three columns: 'Sequence', 'Display Value', and 'Return Value'. It contains three rows:

Sequence	Display Value	Return Value
1	SUDAH DIAMBIL	SUDAH DIAMBIL
2	BELUM DIAMBIL	BELUM DIAMBIL
3		

At the bottom left are '<' and 'Cancel' buttons, and at the bottom right is a 'Create List of Values' button.

Gambar 11.34 Input Value of List

33. Setelah membuat *Value of List* kita ubah type-nya menjadi *Select List*, kemudian pada menu *Value of List* pilih *Shared Components* dan pada pilihan *List of Value* pilih nama *Value of List* yang sudah dibuat sebelumnya seperti pada gambar 11.35 berikut.



Gambar 11.35 Edit Value of List

34. Pada gambar 11.36 berikut adalah tampilan form status setelah diubah menjadi *Combo Box*.

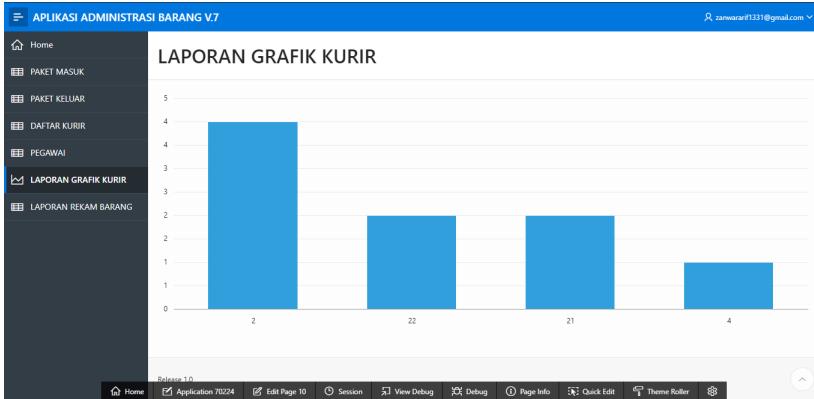
The screenshot shows a 'Barang' form with several input fields:

- Kode Resi
- Tgl Bin Thn
- Kode Via
- Nama Pengirim Paket
- Nama Penerima Paket
- Kode Pegawai
- Status (dropdown menu):
 - SUDAH DIAMBIL
 - BELUM DIAMBIL

At the bottom are 'Cancel' and 'Create' buttons.

Gambar 11.36 Form Pengisian Data Paket Masuk Setelah diedit

35. Berikutnya pada gambar 11.37 adalah tampilan dari menu *Dashboard*. Hanya saja pada tampilan tersebut *Label Value*-nya masih dalam bentuk kode, oleh karena itu kita perlu merubahnya menggunakan *Edit Page*.



Gambar 11.37 Tampilan Dashboard pada Laporan Grafik

36. Setelah masuk ke tampilan *Edit Page*, pergi ke Series yang ada di dalam *content body* pada Laporan Grafik Kurir, kemudian ubah *SQL Query* seperti pada gambar 11.38 berikut, kemudian ganti Label yang akan ditampilkan.

The screenshot shows the Oracle APEX Page Designer interface. On the left, the page structure is outlined with regions like Pre-Rendering, Regions (Breadcrumb Bar), Content Body (LAPORAN GRAFIK KURIR), and Series. In the center, the content body contains a chart placeholder labeled 'LAPORAN GRAFIK KURIR'. On the right, the 'Series' panel is open, showing the 'Source' tab with an SQL query:

```

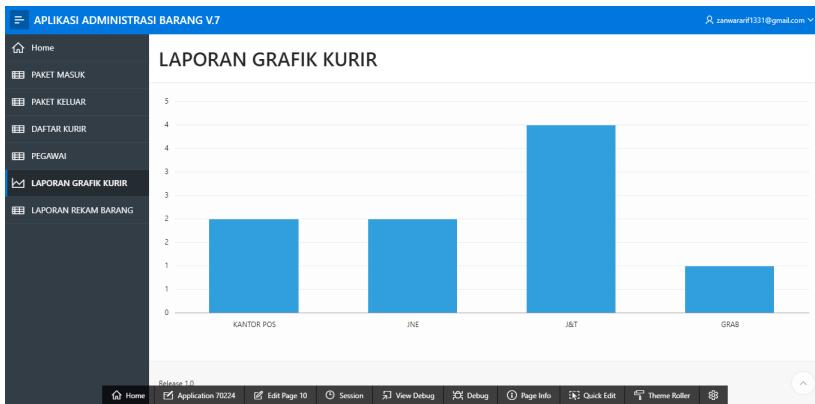
select PAKET_VIA, count(TGL_BLN.Th) value
from TBL_KURIR
inner join TBL_BLN ON BARGA_KODE_VIA =
        KURIR_KODE_VIA
where TGL_BLN.Th is not null
group by PAKET_VIA
order by 1 desc

```

The 'Label' dropdown in the 'Column Mapping' section is set to 'PAKET_VIA'.

Gambar 11.38 Edit Page Dashboard

37. Pada gambar 11.39 berikut adalah tampilan *Dashboard* setelah diedit, dimana akan menampilkan atribut paket_via sebagai Label.



Gambar 11.39 Tampilan Dashboard Hasil Edit

38. Pada gambar 11.40 menampilkan view. Dimana View merupakan hasil perintah *select* yang berupa tabel virtual yang dapat membungkus *query* tertentu. View berfungsi untuk mempermudah dalam pengambilan data pada atribut tertentu.

The screenshot shows a web-based application interface. At the top, there is a blue header bar with the text 'APLIKASI ADMINISTRASI BARANG V.7'. On the right side of the header, there is an email address 'zawarant1331@gmail.com' and a user icon. Below the header, on the left, is a vertical sidebar menu with the following items: Home, PAKET MASUK, PAKET KELUAR, DAFTAR KURIR, PEGAWAI, LAPORAN GRAFIK KURIR, and LAPORAN REKAM BARANG. The 'LAPORAN REKAM BARANG' item is currently selected and highlighted in blue. The main content area has a title 'LAPORAN REKAM BARANG'. Below the title is a search bar with a magnifying glass icon and a dropdown arrow, followed by buttons for 'Go' and 'Actions'. To the right of these buttons is a 'Reset' button. The main area contains a table with the following data:

Kode Resi	Tanggal Masuk	Pegawai Yang Menerima	Tanggal Keluar	Ditambil Oleh	Pegawai Yang Memberi
444-121-111-567	17-JAN-2020	Agius	19-JAN-2020	Sari	Agus
666-09-370-431	02-NOV-2019	Chairul	05-NOV-2019	Zainur Arif Wicaksana	Budi
777-080-361-345	01-NOV-2019	Agus	03-NOV-2019	Anisa Rosalina	Budi
777-080-361-346	03-NOV-2019	Agus	07-NOV-2019	Hanif Weyan Pratama	Budi
888-079-352-250	04-NOV-2019	Budi	06-NOV-2019	Anisa Roselina	Dono
888-079-352-251	05-NOV-2019	Chairul	07-NOV-2019	Zainur Arif Wicaksana	Agus
999-111-232-454	23-JAN-2020	Agius	26-JAN-2020	Ricky	Agus

At the bottom of the table, there is a page number '1 - 7'. Below the table, there is a footer bar with several icons and links: Home, Application 70224, Edit Page 11, Session, View Debug, Debug, Page Info, Quick Edit, Theme Roller, and a gear icon.

Gambar 11.40 User Interface pada View

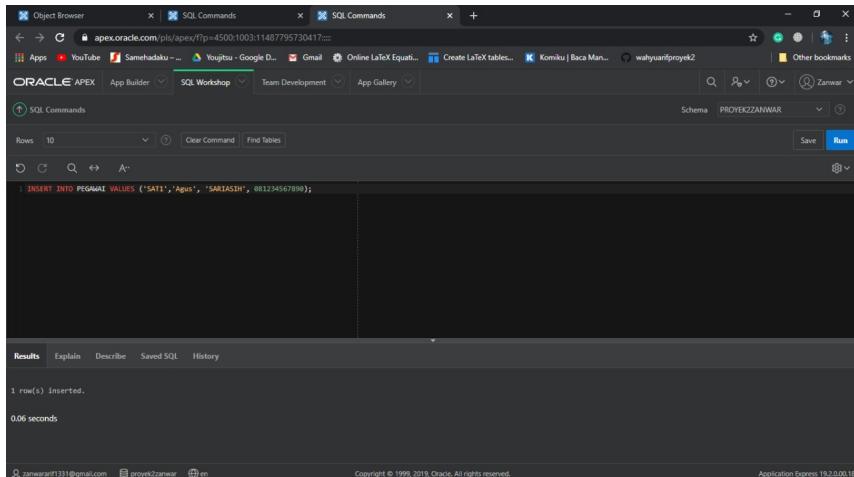
11.2.2 LogBook

1. Pada pembuatan aplikasi administrasi barang ini pasti ditemukan kendala seperti salah mengimputkan data sehingga muncul error pada codingan yang telah dibuat. Seperti halnya pada gambar dibawah berikut ini. Dalam proses pembuatan aplikasi pastikan data-data valid dan sesuai. Pada pembuatan tabel barang seperti pada gambar IV.22 saat dilakukan *running* terjadi kesalahan dalam pengimputan dan data tidak berelasi. Erorr dalam pembuatan tabel barang dibawah ini dapat diatasi seperti gambar IV.23 dengan cara membuat tabel *parent key*.

```
1 CREATE TABLE BARANG (
2   KODE_RESI VARCHAR2(50) PRIMARY KEY,
3   TGL_BLN_THN DATE NOT NULL,
4   KODE_VIA NUMBER NOT NULL,
5   NAMA_PENGIRIM_PAKET VARCHAR2(50) NOT NULL,
6   NAMA_PENERIMA_PAKET VARCHAR2(50) NOT NULL,
7   KODE_PEGAWAI VARCHAR2(50) NOT NULL,
8   STATUS VARCHAR2(50) NOT NULL,
9   DIAMBIL_OLEH VARCHAR2(50),
10  KODE_PEGAWAI1 VARCHAR2(50),
11  TANGGAL_AMBIL DATE,
12  CONSTRAINT FK_VIA1 FOREIGN KEY (KODE_VIA) REFERENCES KURIR,
13  CONSTRAINT FK_PEGAWAI1 FOREIGN KEY (KODE_PEGAWAI) REFERENCES PEGAWAI,
14  CONSTRAINT FK_PEGAWAI2 FOREIGN KEY (KODE_PEGAWAI2) REFERENCES PEGAWAI2
15 );
```

Gambar 11.41 Pembuatan FK Sebelum Membuat Parent Key

2. Pada proses melakukan *insert* ke tabel pegawai hasil *input* nomer telepon dalam bentuk *number* tidak menampilkan angka 0. Untuk mengatasi proses tersebut solusi pertama yang dilakukan dengan menghapus *value* melakukan *modify* pada atribut no hp menjadi *varchar2*, setelah itu mengimputkan ulang datanya. Seperti pada gambar IV.24 merupakan hasil untuk mengatasi error pada tabel ini.



The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. In the SQL editor, the following SQL statement is entered:

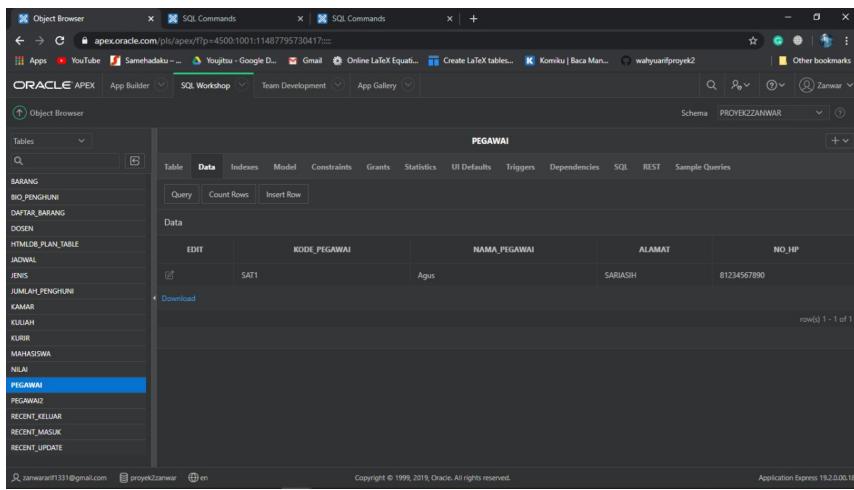
```
1 INSERT INTO PEGAWAI VALUES ('SAT1', 'Agus', 'SARIAISH', 881234567890);
```

After executing the command, the results show:

```
1 row(s) inserted.
```

The execution time is listed as 0.06 seconds. The bottom status bar indicates the user is connected via Gmail and the Application Express version is 19.2.0.0.018.

Gambar 11.42 Tabel Telah Dibuat



The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The left sidebar lists various tables, and the PEGAWAI table is currently selected. The main area displays the PEGAWAI table structure with columns: KODE_PEGAWAI, NAMA_PEGAWAI, ALAMAT, and NO_HP. One row is visible in the data grid:

	KODE_PEGAWAI	NAMA_PEGAWAI	ALAMAT	NO_HP
EDIT	SAT1	Agus	SARIAISH	881234567890

The status bar at the bottom indicates the user is connected via Gmail and the Application Express version is 19.2.0.0.018.

Gambar 11.43 Tabel Pegawai

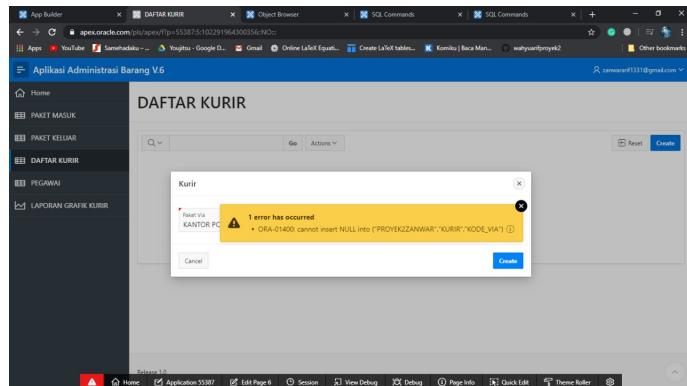
```

1 INSERT INTO PEGAWAI VALUES ('SAT1', 'Agus', 'SARIASIH', 081234567890);
2
3 DELETE FROM PEGAWAI WHERE KODE_PEGAWAI = 'SAT1';
4
5 ALTER TABLE PEGAWAI MODIFY NO_HP VARCHAR(15);
6
7 INSERT INTO PEGAWAI VALUES ('SAT1', 'Agus', 'SARIASIH', '081234567890');
8
9 SELECT * FROM PEGAWAI;
10

```

Gambar 11.44 Tabel Pegawai Telah Dibuat

3. Saat melakukan input ke table kurir melalui aplikasi *apex* tidak menyediakan *field* untuk kode via. Solusi untuk menanganinya adalah membuat *trigger* untuk melakukan auto *increment* dengan *sequence* yang telah dibuat seperti pada gambar IV.26 .Trigger itu sendiri merupakan kode *procedural* yang secara otomatis dijalankan untuk menanggapi perubahan tertentu pada *table* tertentu atau tampilan dalam *database*. *Trigger* dapat didefinisikan untuk menjalankan penrintah sebelum atau setelah eksekusi *DML* (*Data Manipulation Language*) seperti *INSERT*, *UPDATE*, dan *DELETE*. *Trigger* banyak digunakan untuk menjaga integritas informasi pada *database*.

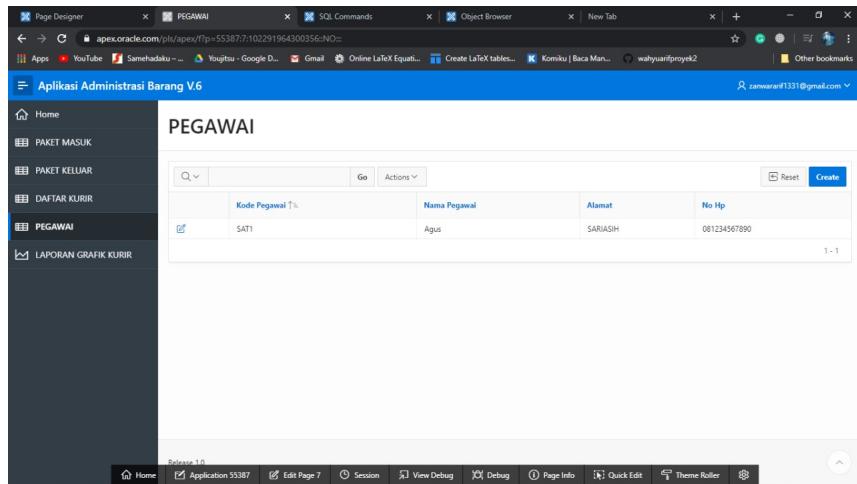


Gambar 11.45 Tabel Kurir Error

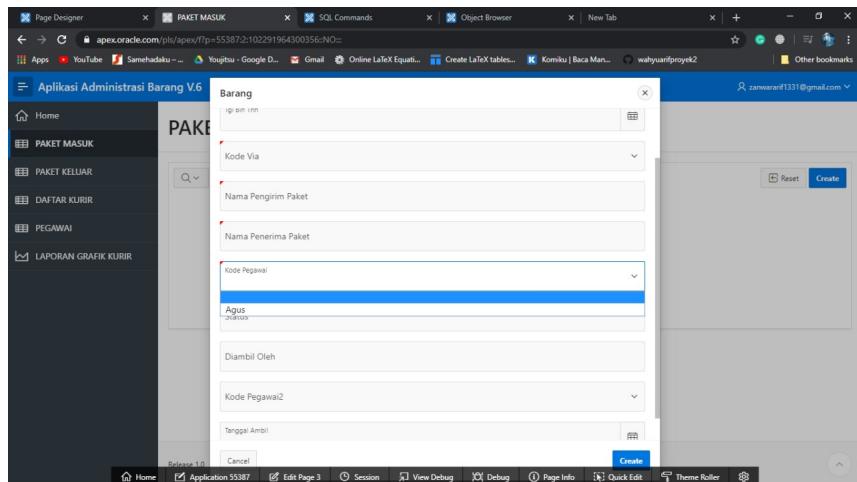
```
1 CREATE SEQUENCE INC_KODE MINVALUE 1 MAXVALUE 999 INCREMENT BY 1;
2
3 CREATE OR REPLACE TRIGGER AUTO_INC1
4 BEFORE INSERT ON KURIR
5 FOR EACH ROW
6 BEGIN
7   SELECT INC_KODE.NEXTVAL
8   INTO :NEW.KODE_VIA
9   FROM dual;
10 END;
11
```

Gambar 11.46 Tabel Kurir Error

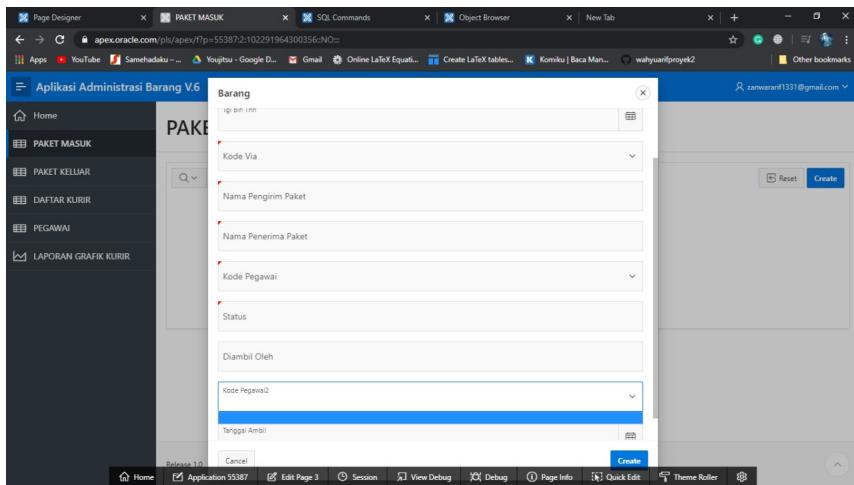
4. Sebelumnya sudah melakukan insert pada table pegawai hanya saja pada atribut dan form yang ada di tabel barang ada kode pegawai2, dimana kode tersebut saling berelasi dengan table pegawai2 sehingga tabel pegawai2 harus memiliki data yang sama dengan tabel pegawai.Selanjutnya adalah membuat Trigger dengan memasukan fungsi *Insert* ke tabel pegawai2, jika ada kegiatan *Insert* ke tabel pegawai seperti pada gambar IV.30.Selanjutnya adalah membuat Trigger dengan melakukan fungsi *Update* ke dalam tabel pegawai2 jika ada kegiatan *Update* ke tabel pegawai seperti pada gambar IV.31. Dalam pembuatan Trigger dengan fungsi *delete* ke tabel pegawai2, jika ada kegiatan *Delete* ke tabel pegawai seperti pada gambar IV.32.Selain itu ada solusi pada proses ini adalah membuat Trigger dengan fungsi *Delete* ke Tabel,pegawai2, jika ada kegiata *insert, update, pegawai menjadi kosong*



Gambar 11.47 Pembuatan Tabel Pegawai



Gambar 11.48 Combo Box



Gambar 11.49 Tabel Barang

```

1 CREATE OR REPLACE TRIGGER TRIGGER_ADD_PEGAWAI
2 AFTER INSERT ON PEGAWAI
3 FOR EACH ROW
4 BEGIN
5   INSERT INTO PEGAWAI2 VALUES (:NEW.KODE_PEGAWAI, :NEW.NAMA_PEGAWAI, :NEW.ALAMAT, :NEW.NO_HP);
6 END;

```

Gambar 11.50 Create Trigger Tabel Pegawai2

```

1 CREATE OR REPLACE TRIGGER TRIGGER_UPDATE_PEGAWAI
2 AFTER UPDATE ON PEGAWAI
3 FOR EACH ROW
4 BEGIN
5   UPDATE PEGAWAI2
6   SET KODE_PEGAWAI2 = :NEW.KODE_PEGAWAI,
7   NAMA_PEGAWAI = :NEW.NAMA_PEGAWAI,
8   ALAMAT = :NEW.ALAMAT,
9   NO_HP = :NEW.NO_HP;
10 END;

```

Gambar 11.51 Create Trigger Update Tabel Pegawai2

```

1 CREATE OR REPLACE TRIGGER TRIGGER_DELETE_PEGAWAI
2 AFTER DELETE ON PEGAWAI
3 FOR EACH ROW
4 BEGIN
5   DELETE FROM PEGAWAI2 WHERE KODE_PEGAWAI2 = :OLD.KODE_PEGAWAI;
6 END;
7
8

```

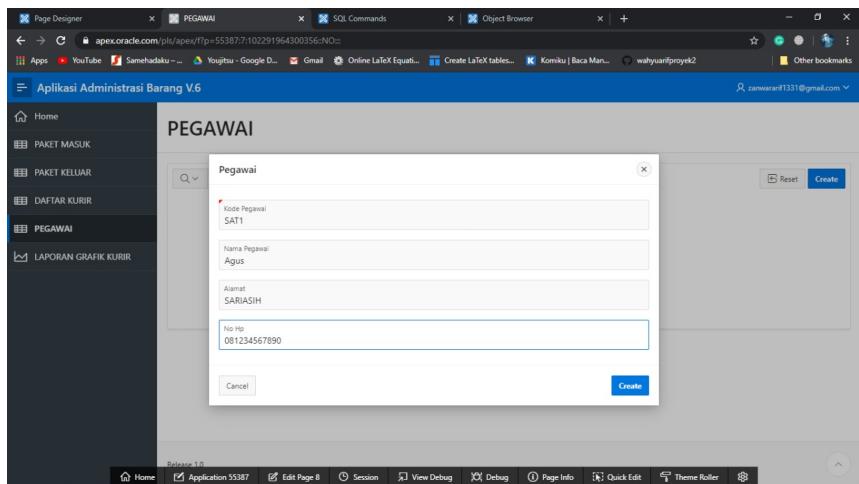
Gambar 11.52 Create Trigger Dengan Fungsi Delete ke Pegawai2

```

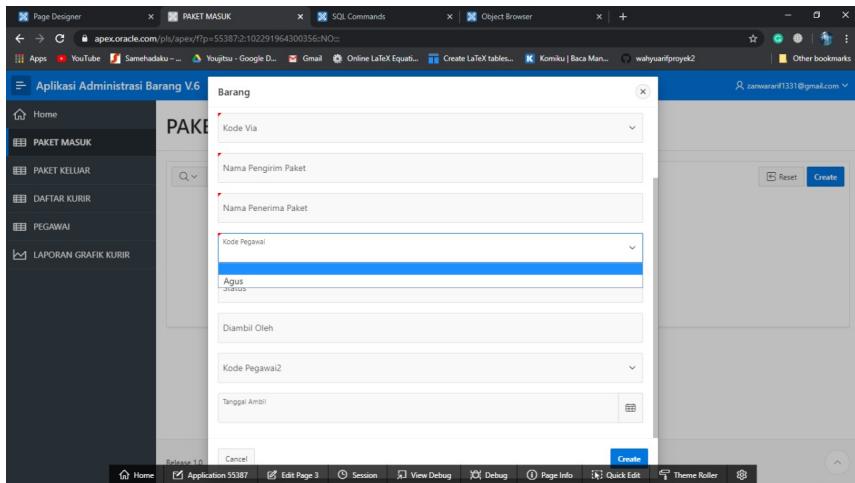
1 CREATE OR REPLACE TRIGGER TRIGGER_PEGAWAI
2 AFTER INSERT OR UPDATE OR DELETE ON PEGAWAI
3 FOR EACH ROW
4 BEGIN
5   IF INSERTING THEN
6     INSERT INTO PEGAWAI2 VALUES (:NEW.KODE_PEGAWAI, :NEW.NAMA_PEGAWAI, :NEW.ALAMAT, :NEW.NO_HP);
7   END IF;
8
9   IF UPDATING THEN
10    UPDATE PEGAWAI2
11    SET KODE_PEGAWAI2 = :NEW.KODE_PEGAWAI,
12        NAMA_PEGAWAI = :NEW.NAMA_PEGAWAI,
13        ALAMAT = :NEW.ALAMAT,
14        NO_HP = :NEW.NO_HP;
15    END IF;
16
17   IF DELETING THEN
18    DELETE FROM PEGAWAI2 WHERE KODE_PEGAWAI2 = :OLD.KODE_PEGAWAI;
19   END IF;
20 END;

```

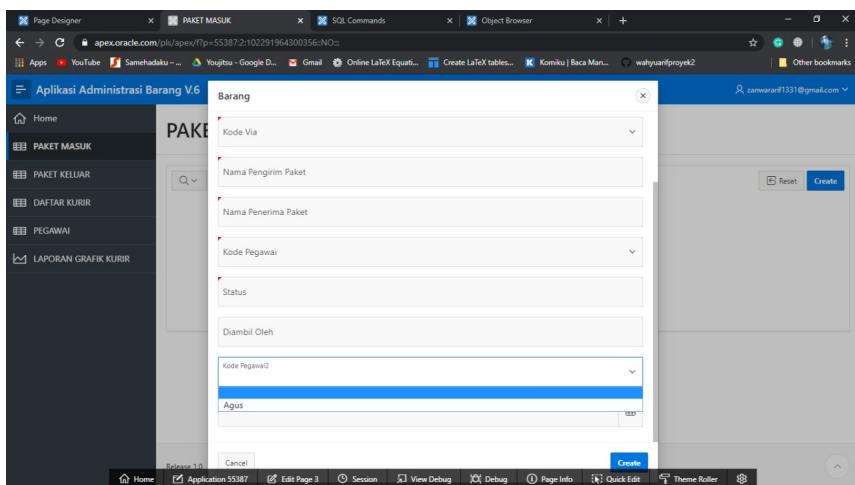
Gambar 11.53 Create Trigger dengan Insert,Update,Delete ke Pegawai



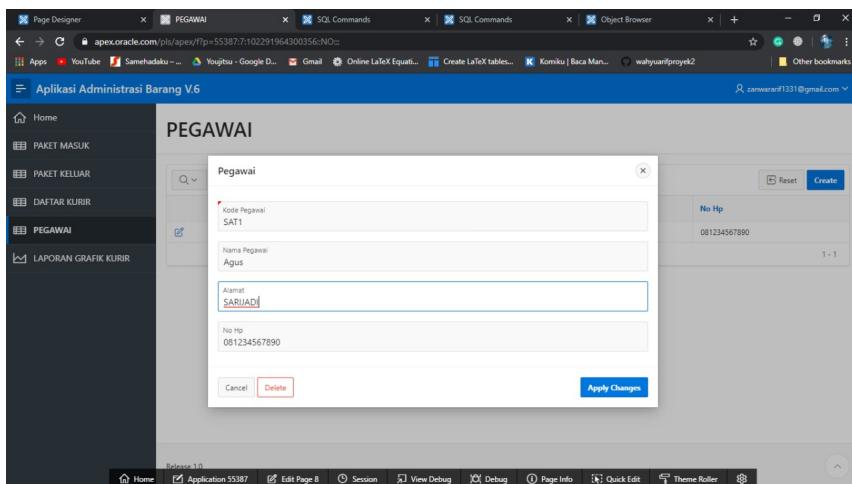
Gambar 11.54 Percobaan Dengan Insert Menggunakan Aplikasi



Gambar 11.55 Percobaan Tabel Pegawai



Gambar 11.56 Percobaan Tabel Pegawai2



Gambar 11.57 Update Pada Field Alamat

```
1 SELECT * FROM PEGAWAI2
```

Results		
KODE_PEGAWAI2	NAMA_PEGAWAI	ALAMAT
SAT1	Agus	SARIASIH

1 rows returned in 0.00 seconds Download

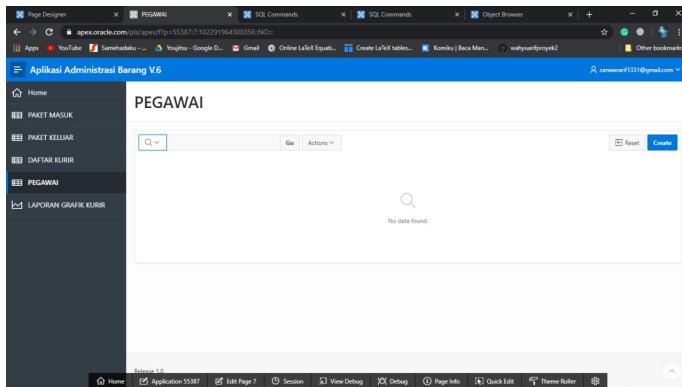
Gambar 11.58 Sebelum Update

```
1 SELECT * FROM PEGAWAI2
```

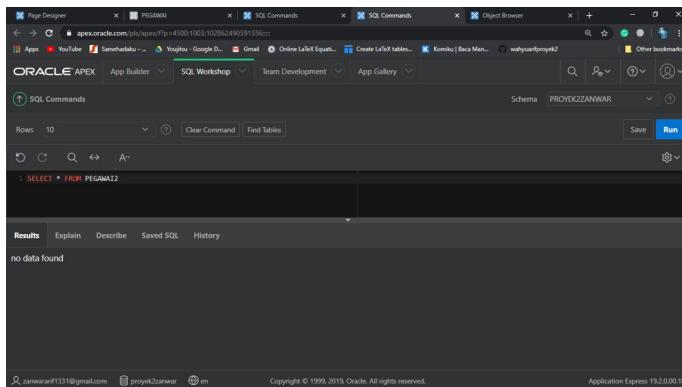
Results			
KODE_PEGAWAI2	NAMA_PEGAWAI	ALAMAT	NO_HP
SAT1	Agus	SARIJADI	81234567890

1 rows returned in 0.00 seconds Download

Gambar 11.59 Setelah Update

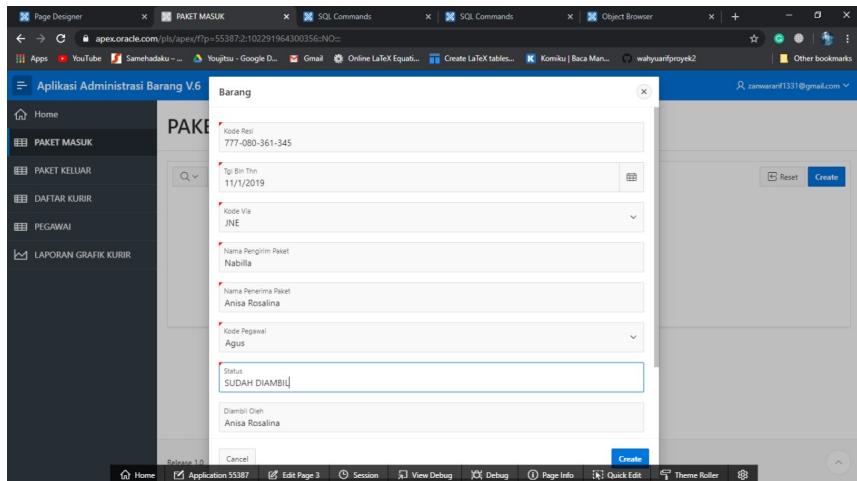


Gambar 11.60 Percobaan Delete

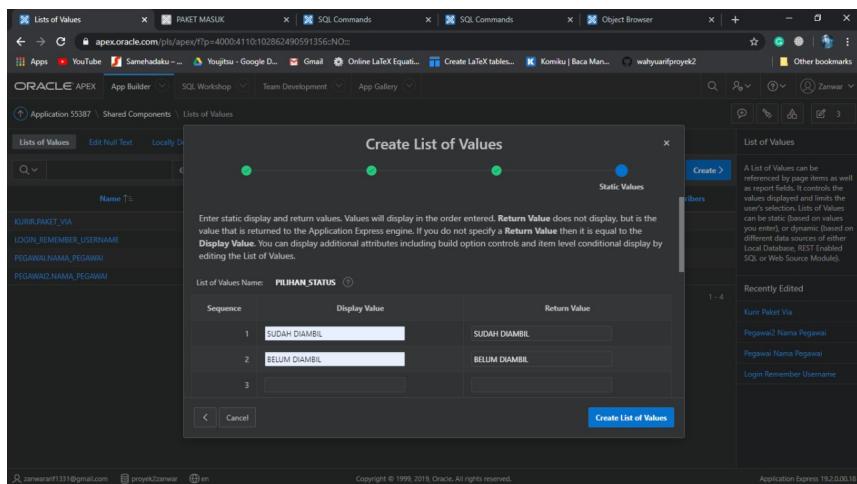


Gambar 11.61 Data Pegawai2 Menjadi Kosong

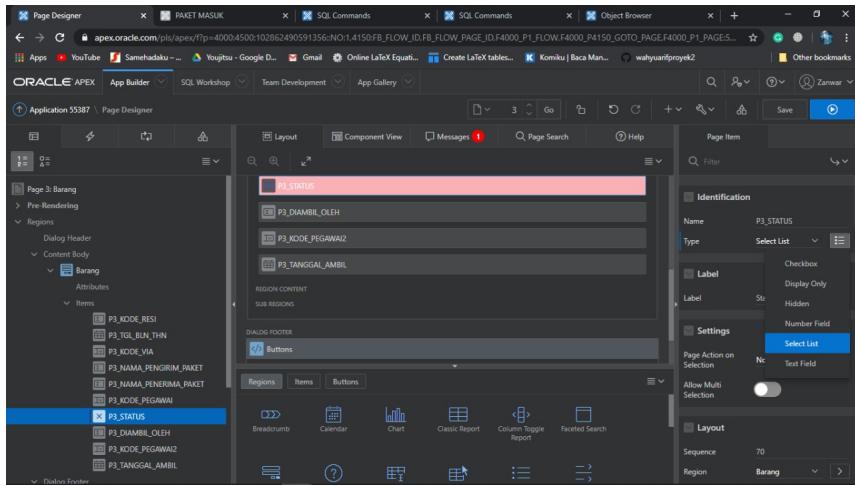
5. Pada *Field status* masih dalam bentuk *text field* untuk menghilangkan kemungkinan terbentuknya banyak data yang berbeda diubah menjadi *combo box*. Pertama membuat *list of value* terlebih dahulu pada gambar IV.43 Selanjutnya adalah membuat mengubah menjadi *select list* seperti gambar IV.44 .Setelah itu mengisi sesuai *value list* pada gambar IV.45 dan yang terakhir merupakan Hasil setelah mengisi *value* seperti hasil dari gambar IV.46



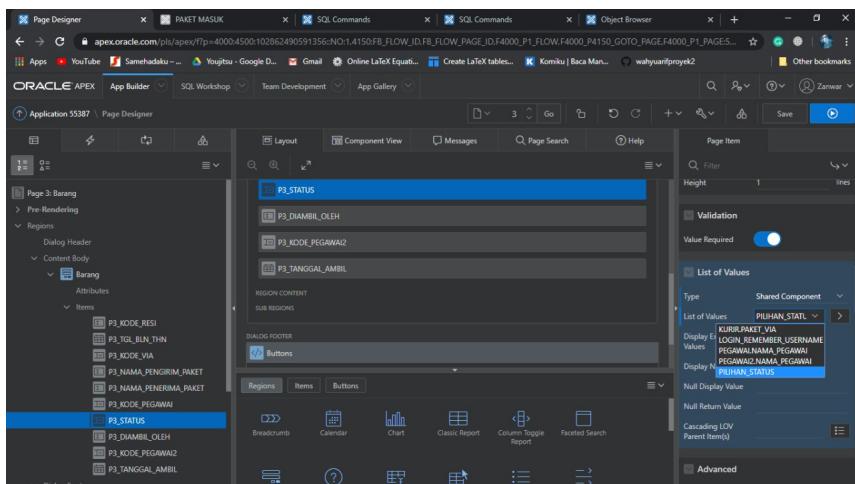
Gambar 11.62 Tampilan Paket Masuk Combo Box



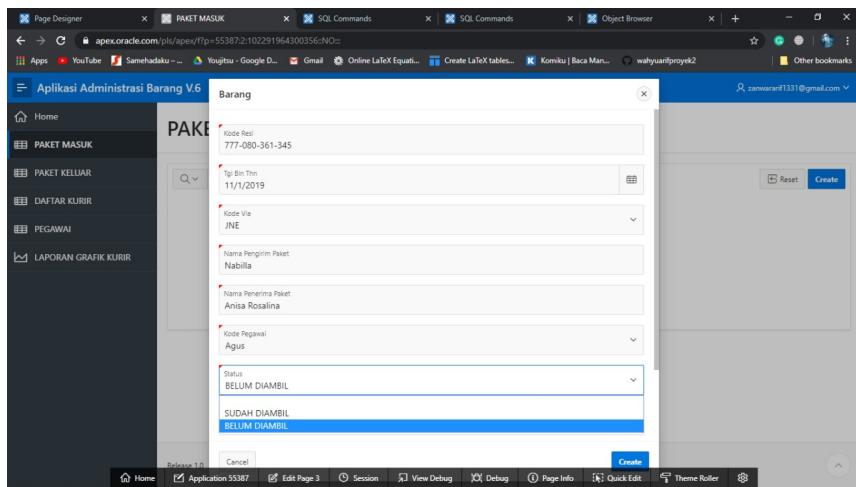
Gambar 11.63 Create list Of Values



Gambar 11.64 Mengubah Menjadi Select List



Gambar 11.65 Mengisi Value Sesuai List

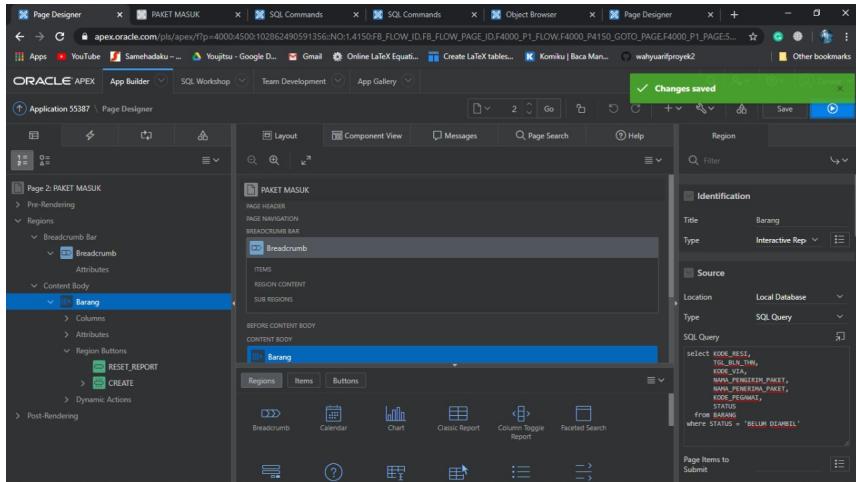


Gambar 11.66 Hasil Mengisi Value

- Pada tampilan paket masuk masih ada beberapa atribut kosong yang ditampilkan dan tujuan dari menu ini hanya akan menampilkan data dengan status "BELUM DIAMBIL". Solusinya adalah dengan cara merubah *query select* pada tabel barang dan menambahkan fungsi *where status = "BELUM DIAMBIL"* seperti pada gambar 11.68. Serta, hasil dari proses ini seperti pada gambar 11.69.

Kode Resi	Tgl Ent Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status	Diambil Oleh	Kode Pegawai2	Tanggal Ambil
777-080-361-345	11/1/2019	JNE	Nabilla	Anisa Rosalina	Agus	BELUM DIAMBIL	-	-	-

Gambar 11.67 Tampilan Paket Masuk



Gambar 11.68 Tampilan Saat Merubah Query

Kode Resi	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status	Diambil Oleh	Kode Pegawai2	Tanggal Ambil
666-091-370-431	11/2/2019	KANTOR POS	Sri Windari Widastuti	Zanwar Arif Wicaksana	Chairul	SUDAH DIAMBIL	Zanwar Arif Wicaksana	SATZ	11/5/2019
777-090-361-345	11/1/2019	JNE	Nabilla	Anisa Rosalina	Agus	BELUM DIAMBIL	-	-	-

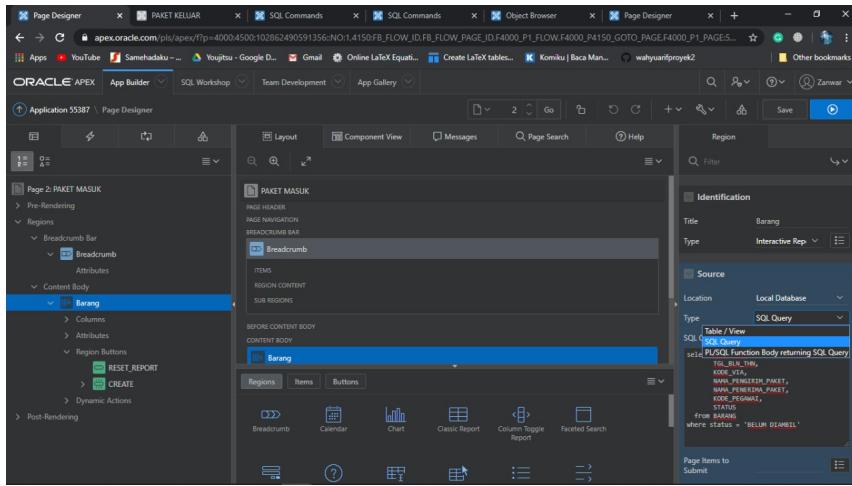
Gambar 11.69 Hasil Sistem Paket Masuk

7. Pada menu paket keluar seperti tertera pada gambar 11.70 merupakan menu tampilan paket keluar. Pada menu ini tujuannya hanya menampilkan data dengan status "SUDAH DIAMBIL". Tambahan untuk sebelumnya merubah *query* merubah type tampilan dari table/view menjadi SQL Query. Untuk mengatasinya adalah merubah *type* tampilan table/view ke SQL Query seperti gambar 11.73. Selanjutnya adalah menambahkan fungsi *where* status "sudah diambil" pada *SQL Query*. Pada gambar 11.74 hasil paket keluar.

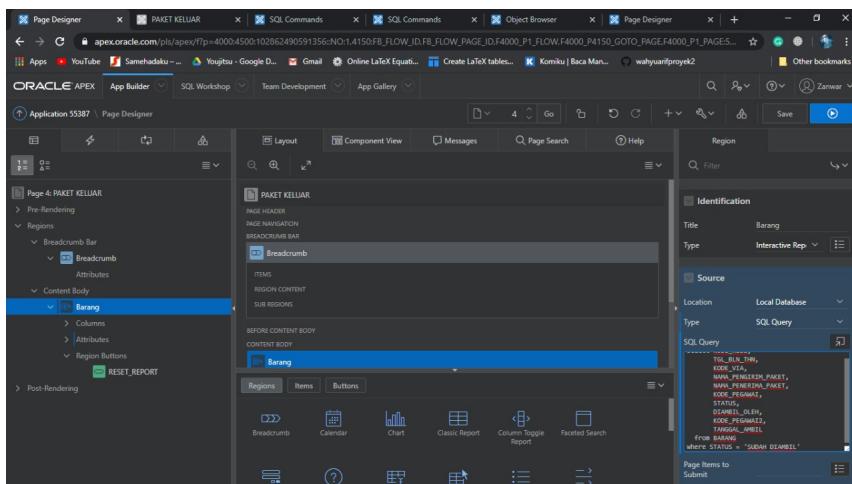
Kode Resi	Tgl Bln Thn	Kode Via	Nama Pengirim	Nama Paket	Nama Penerima	Kode Pegawai	Status	Dambil Oleh	Kode Pegawai	Tanggal Ambil
666-091-370-431	11/2019	KANTOR POS	Sri Windari Widastuti	Zamwar Arif Wicksono	Chainul	SUJAH DIAMBIL	Zamwar Arif Wicksono	Budi	Budi	11/5/2019

Gambar 11.70 Tampilan Menu Paket Keluar

Gambar 11.71 Merubah Tampilan Table/View ke SQL Query



Gambar 11.72 Tampilan Table View Menjadi SQL Query



Gambar 11.73 Menambahkan Fungsi Status

The screenshot shows a web application interface for managing packages. The left sidebar has navigation links: Home, PAKET MASUK, PAKET KELUAR (selected), DAFTAR KURIR, PEGAWAI, and LAPORAN GRAFIK KURIR. The main content area is titled 'PAKET KELUAR' and displays a table of outgoing packages. The table columns are: Kode Resi, Tgl Bln Thn, Kode Via, Nama Pengirim Paket, Nama Penerima Paket, Kode Pegawai, Status, Dambil Oleh, Kode Pegawai2, and Tanggal Ambil. One row is visible in the table:

Kode Resi	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status	Dambil Oleh	Kode Pegawai2	Tanggal Ambil
666-091-370-431	11/2/2019	KANTOR POS	Sri Windari Widiastuti	Zanwar Arif Wicaksono	Chairul	SUDAH DIAMBIL	Zanwar Arif Wicaksono	Budi	11/5/2019

At the bottom of the page, there are various browser tabs and toolbars.

Gambar 11.74 Hasil Paket Keluar

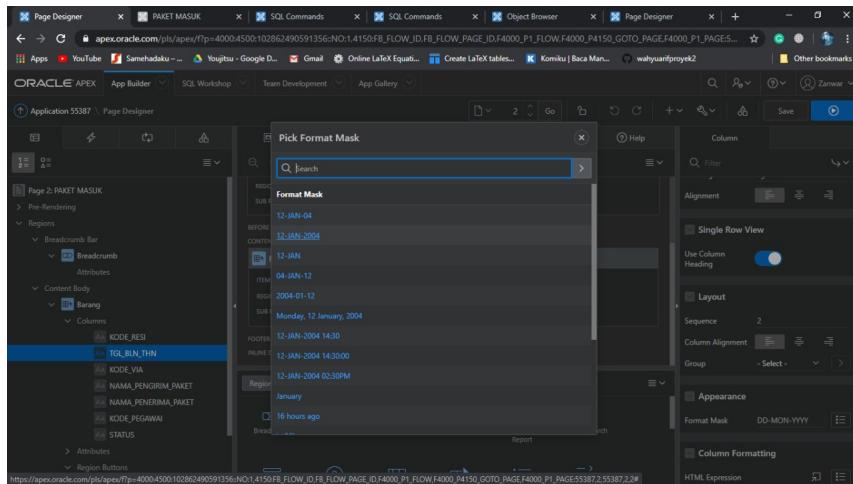
8. Format tanggal pada menut pake masuk masih belum sesuai standar Indonesia seperti pada gambar 11.75. Untuk mengatasi hal tersebut agar dapat dijadikan standar Indonesia dilakukan perubahan *Format Mask* pada menu *Appreance*

The screenshot shows a web application interface for managing packages. The left sidebar has navigation links: Home, PAKET MASUK (selected), PAKET KELUAR, DAFTAR KURIR, PEGAWAI, and LAPORAN GRAFIK KURIR. The main content area is titled 'PAKET MASUK' and displays a table of incoming packages. The table columns are: Kode Resi, Tgl Bln Thn, Kode Via, Nama Pengirim Paket, Nama Penerima Paket, Kode Pegawai, and Status. One row is visible in the table:

Kode Resi	Tgl Bln Thn	Kode Via	Nama Pengirim Paket	Nama Penerima Paket	Kode Pegawai	Status
777-080-361-345	11/1/2019	JNE	Nebila	Anisa Rosalina	Aguus	BELUM DIAMBIL

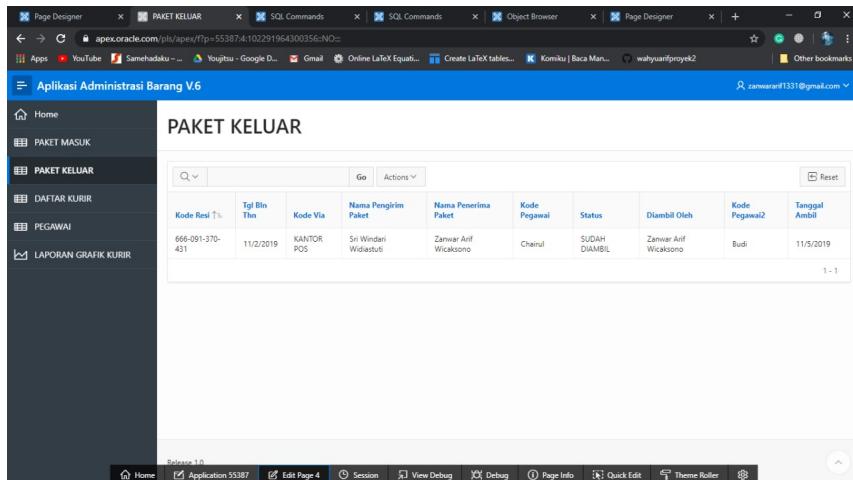
At the bottom of the page, there are various browser tabs and toolbars.

Gambar 11.75 Tampilan Menu Paket Masuk

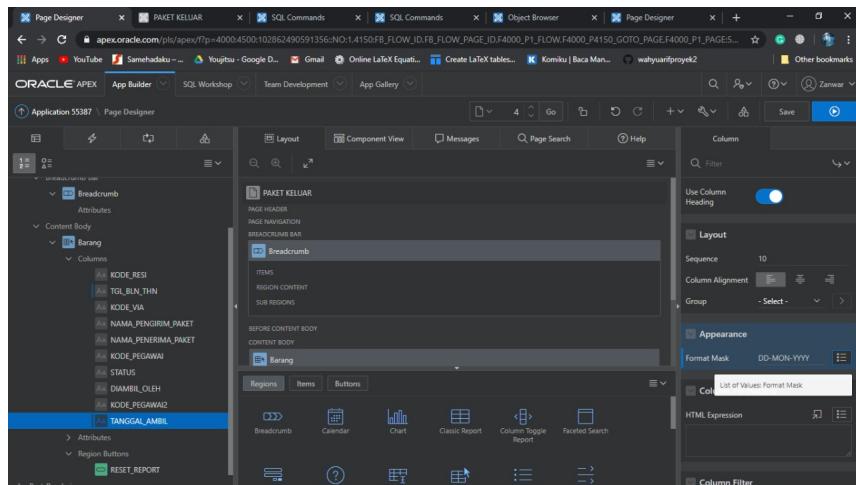


Gambar 11.76 Tampilan Perubahan Format Mask Pada Menu Apprance

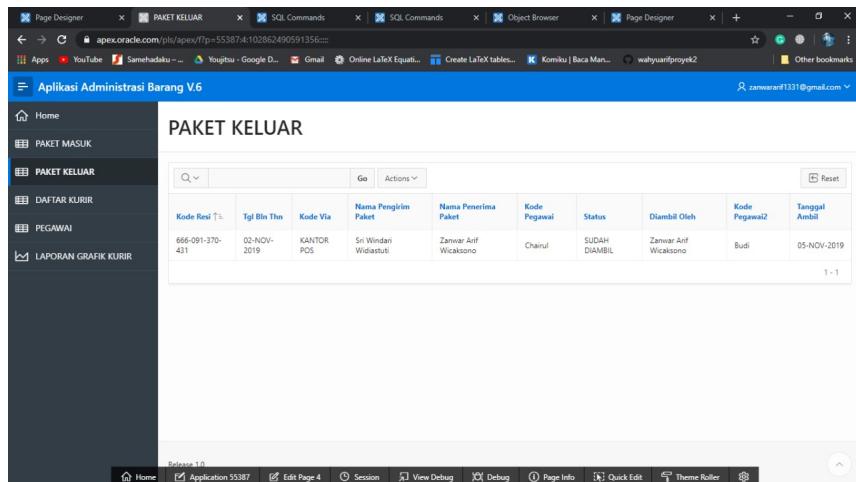
9. Begitu juga dengan menu paket keluar, format penanggalanya masih belum benar seperti pada gambar 11.77 menu paket. Maka dari itu solusi untuk mengatasi hal ini adalah dengan cara menggantinya juga pada *Apparances* dengan memilih atributnya seperti pada gambar 11.78. Dan terakhir adalah hasil dari proses di atas seperti gambar 11.79.



Gambar 11.77 Menu Paket

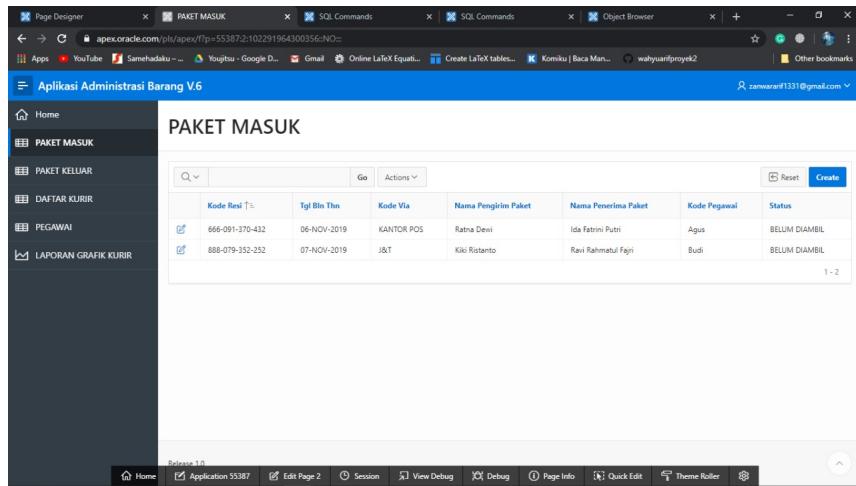


Gambar 11.78 Penggantian pada appearances dengan memilih atribut

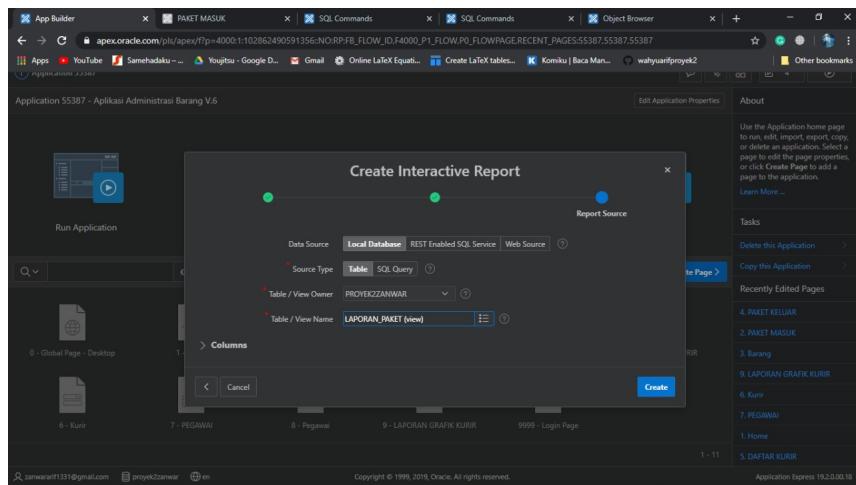


Gambar 11.79 Hasil Paket Keluar

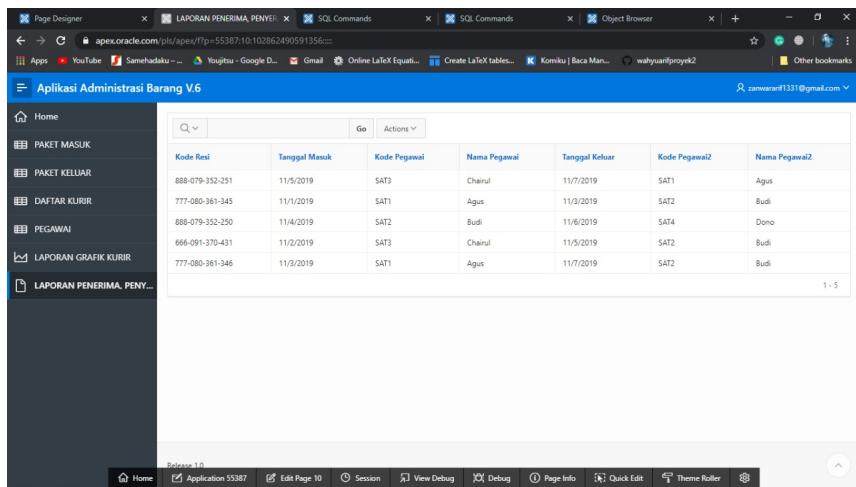
10. Pada aplikasi ini belum menampilkan menu view, maka untuk membuat tampilan menampilkan view



Gambar 11.80 Tampilan Perubahan Format Mask Pada Menu Apprance



Gambar 11.81 Tampilan Perubahan Format Mask Pada Menu Apprance

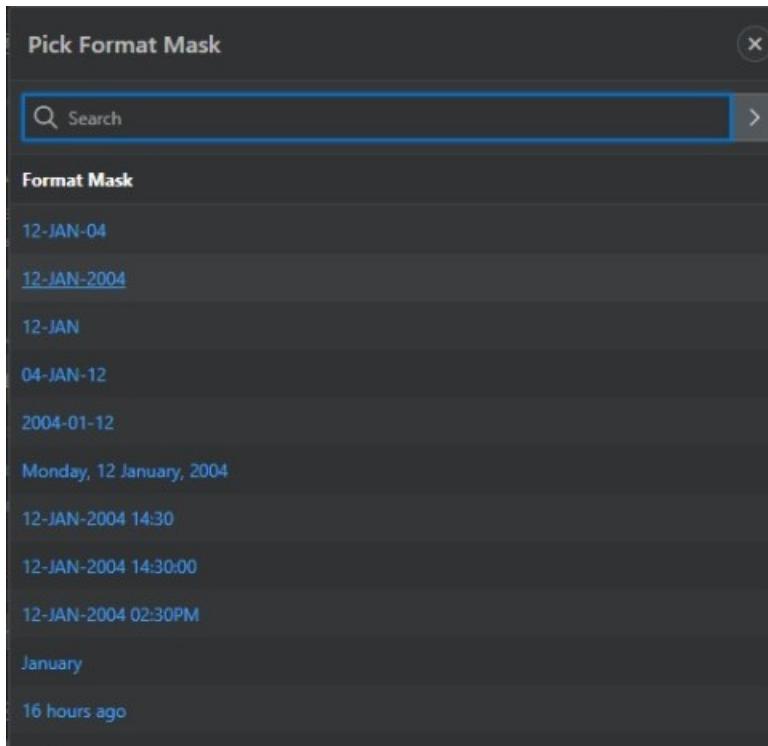


Gambar 11.82 Tampilan Perubahan Format Mask Pada Menu Apprance

- Format tanggal yang ditampilkan belum sesuai ketentuan, solusinya adalah melakukan perubahan pada *format mask* di atribut tanggal

Aplikasi Administrasi Barang V.6																																									
Aplikasi Administrasi Barang V.6																																									
Aplikasi Administrasi Barang V.6																																									
Home																																									
PAKET MASUK	PAKET KELUAR	DAFTAR KURIR	PEGAWAI	LAPORAN GRAFIK KURIR	LAPORAN REKAM BARANG																																				
<table border="1"> <thead> <tr> <th>Kode Resi</th><th>Tanggal Masuk</th><th>Pegawai Yang Menerima</th><th>Tanggal Keluar</th><th>Pegawai Yang Memberi</th><th>Penerima Paket</th></tr> </thead> <tbody> <tr> <td>888-079-352-251</td><td>11/5/2019</td><td>Chairul</td><td>11/7/2019</td><td>Agus</td><td>Zarwar Arif Wicaksono</td></tr> <tr> <td>777-080-361-345</td><td>11/1/2019</td><td>Agus</td><td>11/3/2019</td><td>Budi</td><td>Anisa Rosalina</td></tr> <tr> <td>888-079-352-250</td><td>11/4/2019</td><td>Budi</td><td>11/6/2019</td><td>Dono</td><td>Anisa Rosalina</td></tr> <tr> <td>666-091-370-431</td><td>11/2/2019</td><td>Chairul</td><td>11/3/2019</td><td>Budi</td><td>Zarwar Arif Wicaksono</td></tr> <tr> <td>777-080-361-346</td><td>11/3/2019</td><td>Agus</td><td>11/7/2019</td><td>Budi</td><td>Hanif Wavian Pratama</td></tr> </tbody> </table>						Kode Resi	Tanggal Masuk	Pegawai Yang Menerima	Tanggal Keluar	Pegawai Yang Memberi	Penerima Paket	888-079-352-251	11/5/2019	Chairul	11/7/2019	Agus	Zarwar Arif Wicaksono	777-080-361-345	11/1/2019	Agus	11/3/2019	Budi	Anisa Rosalina	888-079-352-250	11/4/2019	Budi	11/6/2019	Dono	Anisa Rosalina	666-091-370-431	11/2/2019	Chairul	11/3/2019	Budi	Zarwar Arif Wicaksono	777-080-361-346	11/3/2019	Agus	11/7/2019	Budi	Hanif Wavian Pratama
Kode Resi	Tanggal Masuk	Pegawai Yang Menerima	Tanggal Keluar	Pegawai Yang Memberi	Penerima Paket																																				
888-079-352-251	11/5/2019	Chairul	11/7/2019	Agus	Zarwar Arif Wicaksono																																				
777-080-361-345	11/1/2019	Agus	11/3/2019	Budi	Anisa Rosalina																																				
888-079-352-250	11/4/2019	Budi	11/6/2019	Dono	Anisa Rosalina																																				
666-091-370-431	11/2/2019	Chairul	11/3/2019	Budi	Zarwar Arif Wicaksono																																				
777-080-361-346	11/3/2019	Agus	11/7/2019	Budi	Hanif Wavian Pratama																																				
1 - 5																																									

Gambar 11.83 format tanggal



Gambar 11.84 Solusi Format Mask

12. Error saat *compile replace view* penyebab kurangnya atribut, solusinya yaitu melakukan *replace* tanpa menyebutkan atribut

```
1 CREATE OR REPLACE FORCE VIEW "LAPORAN_PAKET" AS
2 SELECT kode_resi, TGL_BLN_THN as tanggal_masuk, pegawai.kode_pegawai,
3 pegawai.nama_pegawai as pegawai_yang_menerima, tanggal_ambil as tanggal_keluar, pegawai2.kode_pegawai2,
4 diambil_oleh, pegawai2.nama_pegawai as pegawai_yang_memberi
5 FROM BARANG
6 INNER JOIN pegawai on barang.kode_pegawai = pegawai.kode_pegawai
7 INNER JOIN pegawai2 on barang.kode_pegawai2 = pegawai2.kode_pegawai2
```

Gambar 11.85 Penanganan error

13. Kurangnya atribut pengambilan barang, melakukan perubahan nama atribut yang ditampilkan pada tabel, yaitu nama pegawai dan nama pegawai2, menjadi pegawai yang menerima dan memberi. Solusinya jadikan 1 yaitu, melakukan perbaikan pada proses pembuatan *view* dengan cara *replace* dengan *sql command* sesuai perubahan yang diinginkan.

Kode Resi	Tanggal Masuk	Kode Pegawai	Tanggal Keluar	Kode Pegawai2	Nama Pegawai	Nama Pegawai2
888-079-352-251	11/5/2019	SAT3	11/7/2019	SAT1	Chairul	Agus
777-080-361-345	11/1/2019	SAT1	11/3/2019	SAT2	Agus	Budi
888-079-352-250	11/4/2019	SAT2	11/6/2019	SAT4	Budi	Dono
666-091-370-431	11/2/2019	SAT3	11/5/2019	SAT2	Chairul	Budi
777-080-361-346	11/3/2019	SAT1	11/7/2019	SAT2	Agus	Budi

Gambar 11.86 Kurangnya Atribut Pengambilan Barang

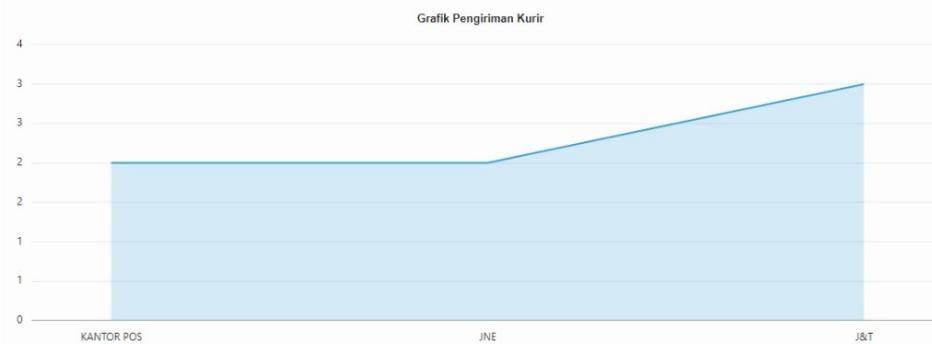
```

CREATE OR REPLACE FORCE VIEW "LAPORAN_PAKET" ("KODE_RESI", "TANGGAL_MASUK", "KODE_PEGAWAI", "NAMA_PEGAWAI", "TANGGAL_KELUAR", "KODE_PEGAWAI2", "DIAMBIL_OLEH",
SELECT kode_resi, TGL_RUM_THN AS tanggal_masuk, pegawai.kode_pegawai,
pegawai.nama_pegawai AS pegawai.yang_menerima, tanggal_ambil AS tanggal_keluar, pegawai2.kode_pegawai2,
diambil_oleh, pegawai2.nama_pegawai AS pegawai.yang_memberi
FROM BARANG
INNER JOIN pegawai ON barang.kode_pegawai = pegawai.kode_pegawai
INNER JOIN pegawai2 ON barang.kode_pegawai2 = pegawai2.kode_pegawai2
  
```

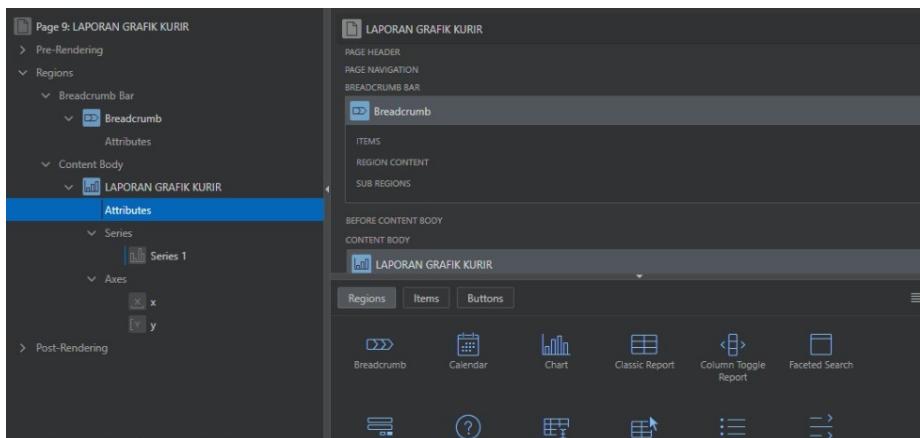
Gambar 11.87 Solusi Proses Pembuatan View

14. Perubahan grafik line ke batang

LAPORAN GRAFIK KURIR



Gambar 11.88 Grafik



Gambar 11.89 Solusi Perubahan Atribut Chart Menjadi Bar

DAFTAR PUSTAKA

1. M. Van Der Plas and M. Van Zoest, *Oracle APEX Cookbook*. Packt Publishing Ltd, 2013.
2. A. Juansyah, “Pembangunan aplikasi child tracker berbasis assisted-global positioning system (a-gps) dengan platform android,” *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 1, no. 1, pp. 2–3, 2015.
3. W. Jonathan and S. Lestari, “Sistem informasi ukm berbasis website pada desa sumber jaya,” *Jurnal Teknologi Informasi dan Bisnis Pengabdian Masyarakat Darmajaya*, vol. 1, no. 1, pp. 1–16, 2015.
4. H. M. Jogiyanto, *Analisis dan Desain (Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis)*. Penerbit Andi, 2017.

