

TUGAS BESAR MANAJEMEN BASIS DATA

Diajukan untuk memenuhi persyaratan kelulusan
Mata Kuliah Manajemen Basis Data

Oleh:
Riadi Satiopa Batubara / 14116136



**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2019**

Daftar Isi

Bab I Pendahuluan	iv
I.1. Latar Belakang	iv
I.2. Studi Literatur	6
I.2.1. Tuning Indexing	6
I.2.2. Setting Configuration DBMS	6
Bab II Hasil dan Pembahasan	7
II.1. Hasil	7
Bab III Kesimpulan	44
III.1. Kesimpulan	44

Bab I Pendahuluan

I.1. Latar Belakang

Pada era globalisasi dalam masyarakat berbasis informasi, sistem basis data merupakan komponen penting pada sebuah sistem informasi manajemen. Suatu sistem besar, yang setiap hari berinteraksi, selalu membutuhkan informasi yang selalu up-to-date dan tepat waktu dalam pengelolaan manajemen data dan informasi.

Basis data dan Sistem Manajemen Basis Data (SMBD) / *Database Management System* (DBMS) dapat menyediakan sarana infrastruktur pada suatu sistem yang dibangun. Sistem manajemen basis data merupakan perangkat lunak yang dapat digunakan untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan basis data.

Tugas dari sistem manajemen basis data adalah menyediakan lingkungan yang nyaman dan efisien untuk penyimpanan dan pengambilan data dari basis data. Pengelolaan manajemen basis data membutuhkan suatu perangkat / tools untuk dapat mengelolanya, sehingga manajemen basis data dapat terus dikelola dan terus ditingkatkan kinerjanya.

Performance tuning merupakan suatu usaha ataupun cara untuk memperbaiki kinerja yang sudah umum bagi perangkat keras dan perangkat lunak komputer yang diharapkan dapat menjaga kestabilan pada suatu sistem yang ada. Maka dari itu pada makalah ini penulis membahas tentang *performance tuning* lebih lanjut sebagai penambahan ilmu dan juga sebagai tugas dari mata kuliah Sistem Basis Data.

I.2. Studi Literatur

I.2.1. Tuning Indexing

Indeks *tuning* adalah bagian *tuning* basis data untuk memilih dan membuat indeks. Tujuan *tuning* indeks adalah untuk mengurangi waktu pemrosesan *query*. Potensi penggunaan indeks di lingkungan dinamis dengan beberapa permintaan sebelumnya yang sulit. *tuning* indeks melibatkan *query* berdasarkan indeks dan indeks dibuat secara otomatis saat itu juga. Tidak diperlukan tindakan eksplisit oleh pengguna basis data untuk *tuning* indeks.

Pembuatan indeks dibuat dengan cara menggunakan beberapa kolom pada *database*. Pada kolom pertama merupakan kunci pencarian yang berisi salinan kunci utama atau kunci kandidat dari tabel. Nilai-nilai tersebut disimpan dalam urutan diurutkan sehingga data yang sesuai dapat diakses dengan cepat. Data berkemungkinan disimpan dalam urutan yang telah diurutkan.

I.2.2. Setting Configuration DBMS

Dalam *tuning* terdapat juga persyaratan yang harus mencatat konfigurasi perangkat keras dan perangkat lunak yang akan dilakukan *tuning*. Hal ini membantu pada saat akan melakukan *tuning*. Perangkat lunak yang dimaksud itu mencakup juga Database Management Server yang digunakan. Tingkat performa dapat dipengaruhi oleh desain pada database, dimana itu termasuk normalisasi dan penyimpanan disk, jumlah table, desain indexes, dan penggunaan DDL serta parameter terkait.

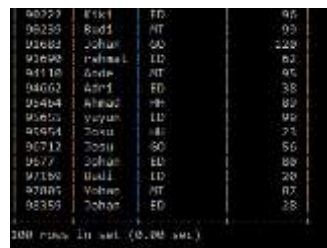
Bab II Hasil dan Pembahasan

II.1. Hasil

Pada saat kondisi advisor = 100, student = 100, section = 200 dan takes = 200. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. Select * From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.



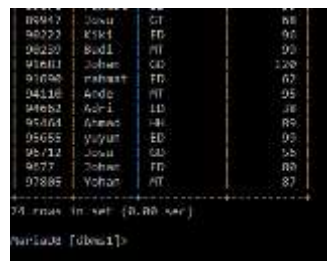
The screenshot shows a table with 4 columns: student_id, first_name, last_name, and tot_cred. The data is as follows:

student_id	first_name	last_name	tot_cred
98723	Kiki	FD	95
98239	Rudi	MT	99
91681	John	GD	120
91698	Robb	LD	62
91119	Ande	MT	95
91662	Arif	FD	38
91684	Alma	MT	89
91655	Yayut	LD	99
91954	Danu	MT	71
91712	Danu	GD	56
91677	Danu	LD	80
91130	Rudi	LD	20
91885	Vahan	MT	87
91359	Danu	FD	28

100 rows in set (0.00 sec)

2. SELECT * FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.



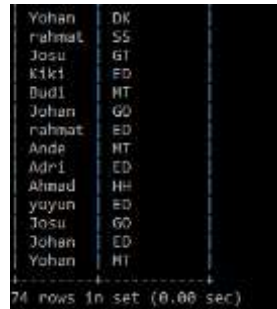
The screenshot shows a table with 4 columns: student_id, first_name, last_name, and tot_cred. The data is as follows:

student_id	first_name	last_name	tot_cred
91954	Danu	MT	71
98723	Kiki	FD	95
98239	Rudi	MT	99
91681	John	GD	120
91698	Robb	FD	62
91119	Ande	MT	95
91684	Arif	LD	38
91684	Alma	MT	89
91655	Yayut	LD	99
91712	Danu	GD	56
91677	Danu	FD	80
91885	Vahan	MT	87

14 rows in set (0.00 sec)

MySQL [db1]:

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`
Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

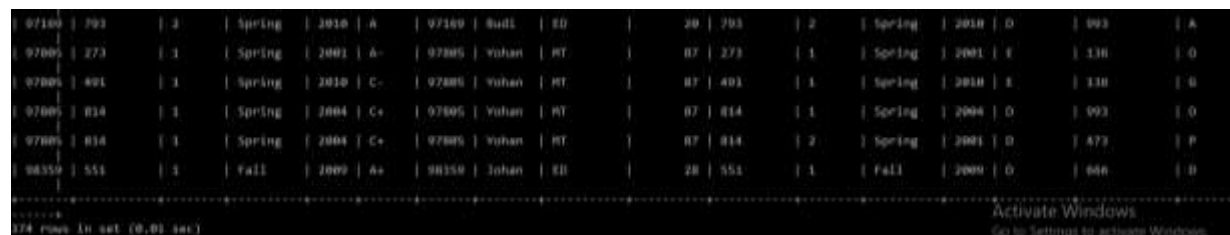


Yohan	DK
rahmat	SS
Josu	GI
Kiki	ED
Budi	MT
Johan	GO
rahmat	ED
Ande	MT
Adri	ED
Ahmad	HM
yayun	ED
Josu	GO
Johan	ED
Yohan	MT

74 rows in set (0.06 sec)

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

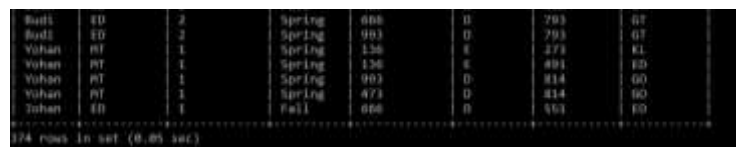


97100	793	2	Spring	2010	A	97100	Budi	ED	20	793	2	Spring	2010	D	993	A
97000	273	1	Spring	2001	A-	97000	Yohan	MT	87	273	1	Spring	2001	E	138	D
97000	491	1	Spring	2010	C-	97000	Yohan	MT	87	491	1	Spring	2010	E	138	B
97000	814	1	Spring	2004	C+	97000	Yohan	MT	87	814	1	Spring	2004	D	993	D
97000	814	1	Spring	2004	C+	97000	Yohan	MT	87	814	2	Spring	2004	D	473	P
98350	551	1	Fall	2009	A+	98350	Johan	ED	28	551	1	Fall	2009	D	046	D

174 rows in set (0.01 sec)

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS
pengambilan,takes.semester,section.room_number,section.building,course.course
_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN
section ON takes.course_id = section.course_id JOIN course ON
section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.



Buti	ID	2	Spring	000	0	793	07
Buti	ID	2	Spring	903	0	793	07
Yohan	BT	1	Spring	130	0	273	01
Yohan	BT	1	Spring	130	0	001	00
Yohan	BT	1	Spring	903	0	814	00
Yohan	BT	1	Spring	873	0	814	00
Yohan	ID	1	Fall	000	0	353	00

.....
124 rows in set (0.85 sec)

Lalu pada file `my.ini` di folder `xampp/mysql/bin`, ganti isinya dengan format berikut untuk melakukan index tuning.

```
#  
  
# FromDual configuration file template for MySQL, Galera Cluster, MariaDB and  
Percona Server  
  
# Location: %MYCNF%  
  
# This template is intended to work with MySQL 5.7 and newer and MariaDB 10.0  
and newer  
  
# Get most recent updated from here:  
# https://www.fromdual.com/mysql-configuration-file-sample  
  
#  
  
[client]  
  
  
port = %PORT% #  
default 3306
```

```
socket                                = %SOCKET%                                # Use
mysql.sock on Ubuntu, conflicts with AppArmor otherwise
```

```
[mysql]
```

```
no_auto_rehash
max_allowed_packet                    = 16M
prompt                               = '\u@\h [\d]> '                                #
'user@host [schema]> '
default_character_set                 = utf8                                        #
Possibly this setting is correct for most recent Linux systems
```

```
[mysqldump]
```

```
max_allowed_packet                    = 16M
```

```
[mysqld_safe]                        #
Becomes sooner or later obsolete with systemd
```

```
open_files_limit                      = 8192                                # You
possibly have to adapt your O/S settings as well
user                                  = mysql
log-error                             =
%INSTANCEDIR%/log/%UNAME%_%INSTANCE%_error.log # Adjust AppArmor
configuration: /etc/apparmor.d/local/usr.sbin.mysqld
```

```
[mysqld]
```

```
# Connection and Thread variables
```

```

port                                = %PORT%                                #
default 3306

socket                              = %SOCKET%                                # Use
mysql.sock on Ubuntu, conflicts with AppArmor otherwise

basedir                             = %BASEDIR%

datadir                             = %DATADIR%

# tmpdir                            = '%INSTANCEDIR%/tmp'
# innodb_tmpdir                      = '%INSTANCEDIR%/tmp'                                #
MySQL 5.7

max_allowed_packet                  = 16M

default_storage_engine              = InnoDB

# explicit_defaults_for_timestamp = 1                                #
MySQL 5.6, test carefully! This can have an impact on application.

# disable_partition_engine_check = true                                #
Since MySQL 5.7.17 to 5.7.20. To get rid of nasty message in error log

# character_set_server              = utf8mb4                                # For
modern applications, default in MySQL 8.0

# collation_server                  = utf8mb4_general_ci

max_connections                      = 151                                # Values
< 1000 are typically good

max_user_connections                = 145                                # Limit
one specific user/application

thread_cache_size                   = 151                                # Up to
max_connections makes sense

# Query Cache (does not exist in MySQL 8.0 any more!)

```

```
# query_cache_type          = 1                # Set
to 0 to avoid global QC Mutex
```

```
# query_cache_size          = 32M              #
Avoid too big (> 128M) QC because of QC clean-up lock!
```

```
# Session variables
```

```
sort_buffer_size           = 2M                # Could
be too big for many small sorts
```

```
tmp_table_size             = 32M              # Make
sure your temporary results do NOT contain BLOB/TEXT attributes
```

```
read_buffer_size           = 128k             # Resist
to change this parameter if you do not know what you are doing
```

```
read_rnd_buffer_size       = 256k             # Resist
to change this parameter if you do not know what you are doing
```

```
join_buffer_size           = 128k             # Resist
to change this parameter if you do not know what you are doing
```

```
# Other buffers and caches
```

```
table_definition_cache     = 1400             # As big
as many tables you have
```

```
table_open_cache           = 2000             #
connections x tables/connection (~2)
```

```
table_open_cache_instances = 16               # New
default in 5.7
```

```
# MySQL error log
```

```
log_error                                =  
%INSTANCEDIR%/log/%UNAME%_%INSTANCE%_error.log  # Adjust AppArmor  
configuration: /etc/apparmor.d/local/usr.sbin.mysqld
```

```
# log_timestamps                        = SYSTEM                                #  
MySQL 5.7, equivalent to old behaviour  
  
log_warnings                            = 2                                # MySQL  
5.6, equivalent to log_error_verbosity = 3  
  
# log_error_verbosity                  = 3                                #  
MySQL 5.7, equivalent to log_warnings = 2, MariaDB does NOT support this!  
  
innodb_print_all_deadlocks              = 1  
  
# wsrep_log_conflicts                  = 1                                # for  
Galera only!
```

Slow Query Log

```
slow_query_log_file                     =  
%INSTANCEDIR%/log/%UNAME%_%INSTANCE%_slow.log  # Adjust AppArmor  
configuration: /etc/apparmor.d/local/usr.sbin.mysqld  
  
slow_query_log                          = 0  
  
log_queries_not_using_indexes           = 0                                #  
Interesting on developer systems!  
  
long_query_time                         = 0.5  
  
min_examined_row_limit                 = 100
```

General Query Log

```
general_log_file                        =  
%INSTANCEDIR%/log/%UNAME%_%INSTANCE%_general.log  # Adjust AppArmor  
configuration: /etc/apparmor.d/local/usr.sbin.mysqld  
  
general_log                             = 0
```

Performance Schema

```

# performance_schema = ON # for
MariaDB 10 releases

performance_schema_consumer_events_statements_history_long = ON # MySQL
5.6/MariaDB 10 and newer

# Binary logging and Replication

server_id = %SERVERID% # Must
be set on MySQL 5.7 and newer if binary log is enabled!

log_bin =
%INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%_binlog # Locate outside of
datadir, adjust AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysqld

# master_verify_checksum = ON #
MySQL 5.6

binlog_cache_size = 1M

binlog_stmt_cache_size = 1M

max_binlog_size = 128M # Make
bigger for high traffic to reduce number of files

sync_binlog = 1 # Set to
0 or higher to increase write performance

expire_logs_days = 5 # We
will survive easter holidays

binlog_format = ROW # Use
MIXED if you want to experience some troubles

# binlog_row_image = MINIMAL #
Since 5.6

# auto_increment_increment = 2 # For
Master/Master set-ups use 2 for both nodes

# auto_increment_offset = 1 # For
Master/Master set-ups use 1 and 2

# Slave variables

```

```

log_slave_updates          = 1                      # Use if
Slave is used for Backup and PiTR

read_only                  = 0                      # Set to
1 to prevent writes on Slave

# super_read_only          = 0                      # Set
to 1 to prevent writes on Slave for users with SUPER privilege. Since 5.7,
not in MariaDB

# skip_slave_start         = 1                      # To
avoid start of Slave thread

# relay_log                = %UNAME%_%INSTANCE%_relay-bin

# relay_log_info_repository = table                #
MySQL 5.6

# master_info_repository   = table                #
MySQL 5.6

# slave_load_tmpdir        = '%INSTANCEDIR%/tmp'

# Crash-safe replication Master

# binlog_checksum          = CRC32                  #
default

# sync_binlog              = 1                      #
default since 5.7.6, but slow!

# innodb_support_xa        = 1                      #
default, depracted since 5.7.10

# Crash-safe replication Slave

# master_info_repository   = TABLE

# relay_log_info_repository = TABLE

# relay_log_recovery       = 1

# sync_relay_log_info       = 1

# relay_log_purge          = 1                      #
default

```

```

# slave_sql_verify_checksum      = 1                                #
default

# GTID replication

# gtid_mode                      = ON                                #
Master and Slave

# enforce_gtid_consistency       = 1                                #
Master and Slave

# log_bin                       =                                     #
%INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%.binlog  # In 5.6 also on Slave

# log_slave_updates              = 1                                # In
5.6 also on Slave

# Security variables

# local_infile                   = 0                                # If
you are security aware

# secure_auth                    = 1                                # If
you are security aware

# sql_mode                      =
TRADITIONAL,ONLY_FULL_GROUP_BY,NO_ENGINE_SUBSTITUTION,STRICT_ALL_TABLES,NO_ZE
RO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER  # Be
careful changing this afterwards

# skip_name_resolve              = 0                                # Set
to 1 if you do not trust your DNS or experience problems

# secure_file_priv               = '%INSTANCEDIR%/tmp'            #
chmod 750, adjust AppArmor configuration:
/etc/apparmor.d/local/usr.sbin.mysqld

# MyISAM variables

```



```

key_buffer_size                = 8M                                # Set to
25 - 33 % of RAM if you still use MyISAM

myisam_recover_options          = 'BACKUP,FORCE'

# disabled_storage_engines      = 'MyISAM,MEMORY'                #
MySQL 5.7, do NOT during/before mysql_upgrade, good for Galera!

# MEMORY variables

max_heap_table_size             = 64M                                # Should
be greater or equal to tmp_table_size

# InnoDB variables

innodb_strict_mode              = ON

# innodb_file_format_check      = 1                                #
Desupported in MySQL 8.0

# innodb_file_format            = Barracuda                        # For
dynamic and compressed InnoDB tables, default in 5.7

innodb_buffer_pool_size         = 128M                                # Go up
to 80% of your available RAM

innodb_buffer_pool_instances    = 8                                # Bigger
if huge InnoDB Buffer Pool or high concurrency

innodb_file_per_table           = 1                                # Is the
recommended way nowadays

# innodb_flush_method           = O_DIRECT                          #
O_DIRECT is sometimes better for direct attached storage

# innodb_write_io_threads       = 8                                # If
you have a strong I/O system or SSD

# innodb_read_io_threads        = 8                                # If
you have a strong I/O system or SSD

# innodb_io_capacity            = 1000                             # If
you have a strong I/O system or SSD

```

```
innodb_flush_log_at_trx_commit = 2                # 1 for
durability, 0 or 2 for performance

innodb_log_buffer_size          = 8M              # Bigger
if innodb_flush_log_at_trx_commit = 0

innodb_log_file_size            = 256M            # Bigger
means more write throughput but longer recovery time
```

```
# Galera specific MySQL parameter
```

```
# default_storage_engine        = InnoDB          # Galera
only works with InnoDB

# innodb_flush_log_at_trx_commit = 2              #
Durability is achieved by committing to the Group

# innodb_autoinc_lock_mode      = 2               # For
parallel applying

# binlog_format                 = row              # Galera
only works with RBR

# query_cache_type              = 0               # Use QC
with Galera only in a Master/Slave set-up

# query_cache_size              = 0
```

```
# WSREP parameter
```

```
# wsrep_on                      = on               # Only
MariaDB >= 10.1

# wsrep_provider                = /usr/lib/galera/libgalera_smm.so #
Location of Galera Plugin on Ubuntu ?

# wsrep_provider                = /usr/lib64/galera-3/libgalera_smm.so #
Location of Galera Plugin on CentOS 7

# wsrep_provider                = none             #
Start mysqld without Galera

# wsrep_provider_options        = 'gcache.size = 1G' #
Depends on you workload, WS kept for IST
```

```

# wsrep_cluster_name          = "My cool Galera Cluster"          # Same
Cluster name for all nodes

# wsrep_cluster_address      = "gcomm://192.168.0.2,192.168.0.3"  #
Start other nodes like this


# wsrep_node_name            = "Node A"                            #
Unique node name

# wsrep_node_address         = 192.168.0.1                        # Our
address where replication is done

# wsrep_node_incoming_address = 10.0.0.1                         # Our
external interface where application comes from

# wsrep_sync_wait            = 1                                  # If
you need really full-synchronous replication (Galera 3.6 and newer)

# wsrep_slave_threads        = 16                                # 4 -
8 per core, not more than wsrep_cert_deps_distance


# wsrep_sst_method           = rsync                              # SST
method (initial full sync): mysqldump, rsync, rsync_wan, xtrabackup-v2

# wsrep_sst_auth             = sst:secret                        #
Username/password for sst user

# wsrep_sst_receive_address  = 192.168.2.1                      # Our
address where to receive SST


# Group Replication parameter


# default_storage_engine     = InnoDB                            #
Group Replication only works with InnoDB

# server_id                  = %SERVERID%                        #
Should be different on all 3 nodes

# log_bin                    =
%INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%_binlog  # Locate outside of datadir,
adjust AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysqld

# binlog_format              = ROW

```

```

# binlog_checksum          = NONE          # not
default!

# gtid_mode                = ON
# enforce_gtid_consistency = ON
# master_info_repository  = TABLE
# relay_log_info_repository = TABLE
# log_slave_updates       = ON

# slave_parallel_workers  =          # 1-
2/core, max. 10
# slave_preserve_commit_order = ON
# slave_parallel_type      = LOGICAL_CLOCK

# transaction_write_set_extraction = XXHASH64

# loose-group_replication_group_name = "$(uuidgen)" # Must
be the same on all nodes
# loose-group_replication_start_on_boot = OFF
# loose-group_replication_local_address = "192.168.0.1"
# loose-group_replication_group_seeds =
"192.168.0.1,192.168.0.2,192.168.0.3" # All nodes of Cluster
# loose-group_replication_bootstrap_group = OFF
# loose-group_replication_single_primary_mode = FALSE # =
multi-primary

```

Lalu jalan kueri seperti sebelumnya :

1. Select * From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

05055	yayun	ED	00
05054	Joan	PH	25
05710	Joan	GD	55
05777	John	ED	80
07166	Rudi	ED	20
07885	Yohan	PH	87
08120	John	ED	20

100 rows in set (0.00 sec)

```
Parikau [dbms1]:
```

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

05047	Joan	GI	60
05222	Kiki	ED	95
05355	Rudi	PH	40
05683	John	GD	120
05686	rahmat	ED	65
05110	Andi	PH	97
04060	Andi	ED	30
05404	Alfred	PH	80
05655	yayun	ED	90
05710	Joan	GD	55
05777	John	ED	80
07885	Yohan	PH	87

74 rows in set (0.00 sec)

```
Parikau [dbms1]:
```

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

John	GD
rahmat	ED
Andi	PH
Andi	ED
Alfred	PH
yayun	ED
Joan	GD
John	ED
Yohan	PH

74 rows in set (0.00 sec)

```
Parikau [dbms1]:
```

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

97109	793	2	Spring	2010	A	97109	Budi	ED	20	793	2	Spring	2010	D	993	A
97805	273	1	Spring	2001	A-	97805	Yohan	MT	07	273	1	Spring	2001	E	136	D
97805	491	3	Spring	2010	C-	97805	Yohan	MT	07	491	1	Spring	2010	F	136	G
97805	814	1	Spring	2004	C+	97805	Yohan	MT	07	814	1	Spring	2004	D	993	D
97805	814	1	Spring	2004	C+	97805	Yohan	MT	07	814	2	Spring	2001	D	473	F
98359	551	1	Fall	2009	A+	98359	Johan	ED	20	551	1	Fall	2009	D	666	D

-----+
174 rows in set (0.00 sec)

Marisa00 [dbms1]>

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

Josue	OD	2	Spring	150	H	232	MT
Josue	OD	2	Spring	86	G	232	MT
Josue	OD	2	Spring	86	G	232	MT
Johan	ED	2	Spring	445	A	269	BR
Johan	ED	2	Spring	310	C	269	BR
Budi	ED	1	Spring	150	H	106	KL
Budi	ED	1	Spring	130	E	273	KL
Budi	ED	1	Spring	130	E	393	ED
Budi	ED	1	Fall	463	B	586	GT
Budi	ED	2	Spring	000	D	793	GT
Budi	ED	2	Spring	903	D	793	GT
Yohan	MT	1	Spring	150	E	273	KL
Yohan	MT	1	Spring	150	E	491	ED
Yohan	MT	1	Spring	903	D	814	GD
Yohan	MT	1	Spring	473	D	814	GD
Johan	ED	1	Fall	666	D	551	ED

-----+
174 rows in set (0.01 sec)

Marisa00 [dbms1]>

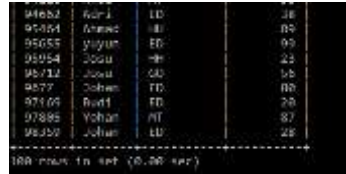
Ulangi percobaan dengan melakukan hal yang sama seperti sebelumnya.

Hasil untuk settings advisor = 200, student = 200, section = 400,takes = 400 seperti berikut.

Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. Select * From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

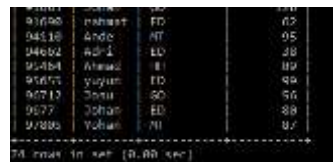


ID	name	dept_name	tot_cred
94682	Adri	ED	38
94684	Amad	ED	39
94655	yuyun	ED	39
94654	Doga	ED	23
94712	Josai	ED	36
94777	Johan	ED	38
97165	Rudi	ED	26
97885	Yohan	ED	37
98332	Johan	ED	28

188 rows in set (0.20 sec)

2. SELECT * FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

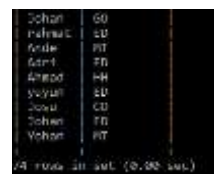


ID	name	dept_name	tot_cred
94682	Adri	ED	38
94684	Amad	ED	39
94655	yuyun	ED	39
94712	Josai	ED	36
94777	Johan	ED	38
97885	Yohan	ED	37

78 rows in set (0.00 sec)

3. SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.



name	dept_name
Adri	ED
Amad	ED
yuyun	ED
Josai	ED
Johan	ED
Yohan	ED

78 rows in set (0.00 sec)

4. SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

07005	273	1	Spring	2001	A-	07005	Yohan	RT	07	273	1	Spring	2001	E	110	0
07005	491	1	Spring	2010	C-	07005	Yohan	RT	07	491	1	Spring	2010	E	116	6
07005	814	1	Spring	2004	C+	07005	Yohan	RT	07	814	1	Spring	2004	D	003	0
07005	814	1	Spring	2004	C+	07005	Yohan	RT	07	814	2	Spring	2004	D	473	P
08110	551	1	Fall	2009	A+	08110	Johan	ED	28	551	1	Fall	2009	D	006	0

174 rows in set (0.00 sec)

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

Rudi	ID	1	Fall	403	0	100	07
Rudi	ID	2	Spring	656	0	791	07
Rudi	ID	3	Spring	903	0	791	07
Yohan	RT	1	Spring	156	8	273	01
Yohan	RT	1	Spring	156	8	491	00
Yohan	RT	1	Spring	903	0	814	00
Yohan	RT	1	Spring	473	0	814	00
Johan	ED	1	Fall	066	0	551	00

174 rows in set (0.00 sec)

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

05005	Yohan	RT	07	273
05054	Johan	ED	28	551
05112	Johan	ED	00	551
05777	Johan	ED	00	551
07105	Rudi	ED	28	551
07885	Yohan	RT	07	273
08110	Johan	ED	28	551

160 rows in set (0.00 sec)

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.



94662	Adi	ED	38
95464	Ahmad	ED	42
95675	yuyun	ED	49
95712	Totus	GO	56
95772	Johan	ED	88
97005	Yohan	ED	117

74 rows in set (0.00 sec)

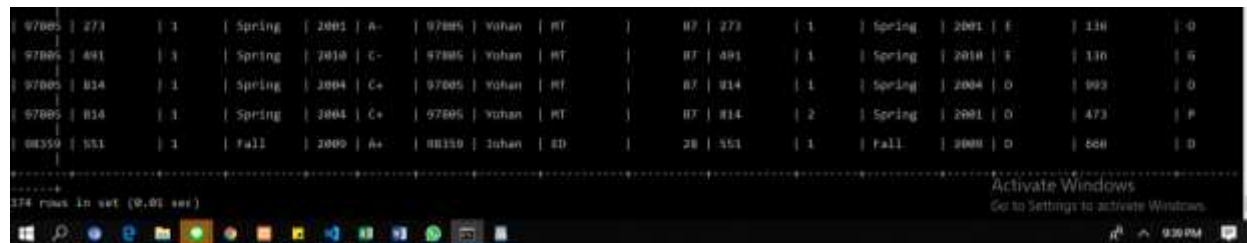
3. `SELECT `name`,`dept_name` FROM student WHERE tot_cred > 30;`
 Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.



Budi	MT
Johan	GO
nahzat	ED
Ande	MT
Adri	ED
Ahmad	MT
yuyun	ED
Totus	GO
Johan	ED
Yohan	MT

74 rows in set (0.00 sec)

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`
 Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.



97005	273	1	Spring	2001	A-	97005	Yohan	MT	87	273	1	Spring	2001	E	118	0
97005	491	1	Spring	2010	C-	97005	Yohan	MT	87	491	1	Spring	2010	E	110	5
97005	814	1	Spring	2004	C+	97005	Yohan	MT	87	814	1	Spring	2004	D	993	0
97005	814	1	Spring	2004	C+	97005	Yohan	MT	87	814	2	Spring	2001	D	473	P
00359	551	1	Fall	2000	A+	00359	Johan	ED	28	551	1	Fall	2000	D	668	0

374 rows in set (0.01 sec)

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`
 Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

Rudi	ED	1	Spring	156	0	100	KL
Rudi	ED	1	Spring	138	8	273	KL
Rudi	ED	1	Spring	138	8	303	ED
Rudi	ED	1	Fall	483	8	586	OT
Rudi	ED	2	Spring	666	0	703	OT
Rudi	ED	2	Spring	993	0	793	OT
Yohan	PT	1	Spring	138	8	273	KL
Yohan	PT	1	Spring	138	8	491	ED
Yohan	PT	1	Spring	993	0	814	GO
Yohan	PT	1	Spring	473	0	814	GO
Johan	ED	1	Fall	666	0	553	ED

104 rows in set (0.00 sec)

Hasil untuk settings advisor = 500, student = 500, section = 1000,takes = 1000 seperti berikut. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. Select * From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

95655	yoyun	ED	95
95954	Jogu	PT	23
95712	Jogu	GO	55
9577	Johan	ED	08
97166	Rudi	ED	29
97886	Yohan	PT	87
95490	Johan	ED	26

100 rows in set (0.00 sec)

2. SELECT * FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

94118	Ande	PT	95
94662	Rudi	ED	28
95954	Rudi	ED	08
95655	yoyun	ED	95
95712	Jogu	GO	55
9577	Johan	ED	89
97005	Yohan	PT	07

74 rows in set (0.00 sec)

mysql> [dbms1] >

3. SELECT `name`,`dept_name` FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

Ande	PT
Rudi	ED
Rudi	ED
yoyun	ED
Jogu	GO
Johan	ED
Yohan	PT

74 rows in set (0.00 sec)

mysql> [dbms1] >

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

```

mysql> SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 97805 | 491 | 1 | Spring | 2010 | C- | 97805 | Yohan | MT | 07 | 491 | 1 | Spring | 2010 | E | 118 | 6 |
| 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | MT | 07 | 814 | 1 | Spring | 2004 | D | 993 | 0 |
| 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | MT | 07 | 814 | 2 | Spring | 2004 | D | 473 | 8 |
| 98110 | 551 | 1 | Fall | 2009 | A+ | 98110 | Iuhan | ED | 28 | 551 | 1 | Fall | 2009 | D | 608 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
174 rows in set (0.00 sec)

MariaDB [dbname]>

```

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

```

mysql> SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Budi | ID | 1 | Fall | 483 | D | 588 | GT |
| Budi | ID | 2 | Spring | 668 | D | 701 | GT |
| Budi | ID | 2 | Spring | 983 | D | 791 | GT |
| Yohan | MT | 1 | Spring | 136 | E | 273 | KL |
| Yohan | MT | 1 | Spring | 136 | E | 401 | ED |
| Yohan | MT | 1 | Spring | 993 | D | 814 | GO |
| Yohan | MT | 1 | Spring | 673 | D | 814 | GO |
| Iuhan | ED | 1 | Fall | 666 | D | 551 | ED |
+-----+-----+-----+-----+-----+-----+-----+-----+
174 rows in set (0.00 sec)

MariaDB [dbname]>

```

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

Rudi	ED	1	Spring	136	E	393	ED
Rudi	ED	1	Fall	463	B	586	ED
Rudi	ED	2	Spring	666	D	793	ED
Rudi	ED	2	Spring	993	D	793	ED
Yohan	ED	1	Spring	136	E	273	ED
Yohan	ED	1	Spring	136	E	491	ED
Yohan	ED	1	Spring	983	D	814	ED
Yohan	ED	1	Spring	473	D	814	ED
Johan	ED	1	Fall	666	D	551	ED

178 rows in set (0.01 sec)
MariaDB [dbms1]>

Hasil untuk settings advisor = 700, student = 700, section = 20000,takes = 20000 seperti berikut. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

95954	Yohan	ED	89
95954	Yohan	ED	28
95712	Yohan	ED	55
95777	Yohan	ED	88
97166	Rudi	ED	39
97885	Yohan	ED	87
98359	Yohan	ED	28

6 rows in set (0.06 sec)

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

95118	Rudi	ED	95
94684	Rudi	ED	39
95954	Yohan	ED	89
95712	Yohan	ED	55
95777	Yohan	ED	88
97885	Yohan	ED	87

6 rows in set (0.00 sec)
MariaDB [dbms1]>

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`

```

code    PT
Adele   ED
Adele   ED
Adele   ED
yeyun   TD
Tosca   GO
Johan   ED
Yohan   RI

```

74 rows in set (0.00 sec)

```

MariaDB [(db1)]>

```

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section
ON takes.course_id = section.course_id`

```

97805 | 491 | 1 | Spring | 2010 | C- | 97805 | Yohan | RT | 07 | 491 | 1 | Spring | 2010 | F | 110 | 6
97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | RT | 07 | 814 | 1 | Spring | 2004 | D | 903 | 8
97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | RT | 07 | 814 | 2 | Spring | 2001 | D | 473 | 9
08350 | 551 | 1 | Fall | 2009 | A+ | 08350 | Zuhay | ED | 28 | 551 | 1 | Fall | 2009 | D | 608 | 10
-----+-----
174 rows in set (0.00 sec)

```

5. SELECT student.`name`,student.dept_name,takes.sec_id AS

```
Rudi ED 1 Fall 483 B 586 G7  
Rudi ED 2 Spring 689 D 793 G7  
Rudi IT 3 Spring 993 D 793 G7  
Yohan RT 1 Spring 136 E 273 KL  
Yohan RT 1 Spring 136 E 401 EO  
Yohan RT 1 Spring 993 D 814 GO  
Yohan RT 1 Spring 473 D 814 GO  
Yohan ED 1 Fall 666 F 551 EO
```

```
374 rows in set (0.00 sec)
```

```
MariaDB [dtnet]>
```

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. Select * From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.



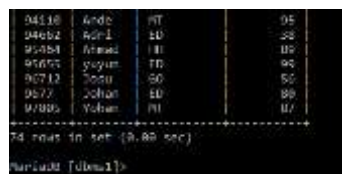
```
mysql> SELECT * FROM student
```

95054	yayan	TI	85
95054	Yogi	TI	23
95112	Johan	SI	55
95777	Johan	TI	88
97105	Rudi	TI	30
97885	Yohan	TI	87
98359	Johan	TI	20

168 rows in set (0.00 sec)

2. SELECT * FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.



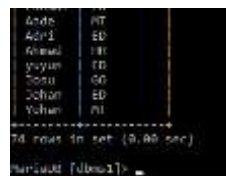
```
mysql> SELECT * FROM student WHERE tot_cred > 30;
```

95110	Ande	TI	95
94682	Rodi	TI	38
95054	Armas	TI	15
95054	yayan	TI	85
95712	Yogi	SI	50
95777	Johan	TI	88
97885	Yohan	TI	87

74 rows in set (0.00 sec)

3. SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.



```
mysql> SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;
```

Ande	TI
Rodi	TI
Armas	TI
yayan	TI
Yogi	SI
Johan	TI
Yohan	TI

74 rows in set (0.00 sec)

4. SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

07805	273	1	Spring	2001	A-	07805	Yohan	RT	87	273	1	Spring	2001	1	138	0
07805	401	1	Spring	2010	C-	07805	Yohan	RT	87	401	1	Spring	2010	1	138	6
07805	814	1	Spring	2004	C+	07805	Yohan	RT	87	814	1	Spring	2004	0	093	0
07805	814	1	Spring	2004	C+	07805	Yohan	RT	87	814	2	Spring	2001	0	473	P
08350	551	1	Fall	2009	A+	08350	Johan	ED	28	551	1	Fall	2009	0	608	0

174 rows in set (0.00 sec)

mysql> [dheul]>

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

Budi	ID	1	Fall	401	8	500	GT
Budi	ID	2	Spring	060	0	701	GT
Budi	ID	2	Spring	983	0	701	GT
Yohan	RT	1	Spring	136	E	273	KL
Yohan	RT	1	Spring	136	E	401	ED
Yohan	RT	1	Spring	983	0	814	GO
Yohan	RT	1	Spring	473	0	814	GO
Johan	ED	1	Fall	066	0	551	ED

174 rows in set (0.00 sec)

mysql> [dheul]>

Hasil untuk settings advisor = 1000, student = 1000, section = 1000000,takes = 1000000 seperti berikut. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

08005	Yohan	RT	87
08054	Yohan	RT	28
08112	Yohan	RT	13
08177	Yohan	RT	10
08186	Budi	RT	28
08285	Yohan	RT	87
08350	Johan	ED	28

100 rows in set (0.00 sec)

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

```

2451.88  kudu      RT      95
2456.2   kudu      IO      28
25.384   kudu      RT      15
256.77   kudu      IO      94
267.12   kudu      IO      93
267.7    kudu      IO      89
268.95   kudu      RT      87
-----+-----
74 rows in set (0.89 sec)

mysql> describe t;

```

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

```

+-----+-----+
| Ande | PT |
| Adri | ED |
| Almad | EE |
| yuyun | TD |
| Jossu | GO |
| Johan | ED |
| Yohan | PI |
+-----+-----+
74 rows in set (0.00 sec)

mysql>

```

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

```

97005 | 491 | 1 | Spring | 2010 | C- | 97005 | Yohan | HT | 07 | 491 | 1 | Spring | 2010 | E | 130 | 0
97005 | 814 | 1 | Spring | 2004 | C+ | 97005 | Yohan | HT | 07 | 814 | 1 | Spring | 2004 | O | 992 | 0
97005 | 814 | 1 | Spring | 2004 | C+ | 97005 | Yohan | HT | 07 | 814 | 2 | Spring | 2001 | D | 473 | P
98350 | 551 | 1 | Fall | 2000 | A+ | 98350 | Zohan | ED | 20 | 551 | 1 | Fall | 2000 | D | 000 | D
.....
174 rows in set (0.01 sec)

```

- ```
5. SELECT student.`name`,student.dept_name,takes.sec_id AS
pengambilan,takes.semester,section.room_number,section.building,course.course
_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN
section ON takes.course_id = section.course_id JOIN course ON
section.course_id = course.course_id
```

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

|       |    |   |        |     |   |     |    |
|-------|----|---|--------|-----|---|-----|----|
| Budi  | ID | 1 | Spring | 136 | 1 | 391 | ED |
| Budi  | ID | 1 | Fall   | 463 | 8 | 386 | OT |
| Budi  | ID | 2 | Spring | 666 | 0 | 791 | OT |
| Budi  | ID | 2 | Spring | 903 | 0 | 791 | OT |
| Yohan | MT | 1 | Spring | 136 | E | 273 | KL |
| Yohan | MT | 1 | Spring | 136 | E | 401 | ED |
| Yohan | MT | 1 | Spring | 903 | 0 | 814 | GO |
| Yohan | MT | 1 | Spring | 473 | 0 | 814 | GO |
| Johan | ID | 1 | Fall   | 666 | 0 | 551 | ED |

174 rows in set (0.01 sec)

hariadi [dbm1]>

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. Select \* From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 95954 | Yohan | MT | 86 |
| 95954 | Yohan | MT | 28 |
| 95712 | Johan | GO | 55 |
| 95777 | Johan | ED | 89 |
| 97165 | Budi  | ED | 39 |
| 97895 | Yohan | MT | 87 |
| 98359 | Johan | ID | 26 |

169 rows in set (0.06 sec)

2. SELECT \* FROM student WHERE tot\_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 95110 | Budi  | MT | 95 |
| 94662 | Budi  | ID | 39 |
| 95364 | Budi  | ID | 85 |
| 95655 | Yohan | ED | 94 |
| 95712 | Johan | GO | 55 |
| 95777 | Johan | ED | 89 |
| 97895 | Yohan | MT | 87 |

74 rows in set (0.09 sec)

hariadi [dbm1]>

3. SELECT `name`, `dept\_name` FROM student WHERE tot\_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

```

 name dept_name sec_id
 ---- -
Gadi PT
Adi ED
Gadi PT
Yayan PT
Soni GT
Johan ED
Yohan PT

74 rows in set (0.00 sec)

MariaDB [dbw1]>

```

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

```

 97805 | 273 | 1 | Spring | 2001 | A- | 97805 | Yohan | PT | 87 | 273 | 1 | Spring | 2001 | 1 | 130 | 0
 97805 | 401 | 1 | Spring | 2010 | C- | 97805 | Yohan | PT | 87 | 401 | 1 | Spring | 2010 | 1 | 130 | 0
 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | PT | 87 | 814 | 1 | Spring | 2004 | 0 | 093 | 0
 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | PT | 87 | 814 | 2 | Spring | 2001 | 0 | 473 | P
 08150 | 551 | 1 | Fall | 2000 | A+ | 08150 | Johan | ED | 28 | 551 | 1 | Fall | 2000 | 0 | 600 | 0

174 rows in set (0.00 sec)

MariaDB [dbw1]>

```

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

```

 name ID semester sec_id building room_no dept_name
 ---- -
Budi ID 2 Spring 666 D 701 GT
Budi ID 2 Spring 993 D 701 GT
Yohan PT 1 Spring 136 E 273 KL
Yohan PT 1 Spring 136 E 401 ED
Yohan PT 1 Spring 093 D 814 GO
Yohan PT 1 Spring 473 D 814 GO
Johan ED 1 Fall 666 D 551 ED

174 rows in set (0.00 sec)

MariaDB [dbw1]>

```

Hasil untuk settings advisor = 1800, student = 1800, section = 180000,takes = 180000 seperti berikut. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

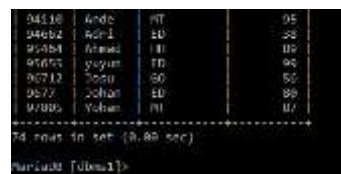


100 rows in set (0.06 sec)

|       |       |    |    |
|-------|-------|----|----|
| 95954 | yayun | IT | 88 |
| 95954 | Yohan | IT | 23 |
| 95712 | Johan | SD | 55 |
| 95777 | Johan | TD | 88 |
| 97165 | Budi  | TD | 38 |
| 97885 | Yohan | MT | 87 |
| 98359 | Johan | TD | 26 |

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

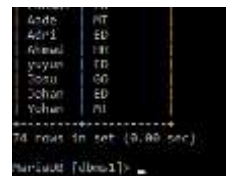


74 rows in set (0.09 sec)

|       |       |    |    |
|-------|-------|----|----|
| 94118 | Andi  | MT | 95 |
| 94682 | Adri  | ED | 38 |
| 95884 | Amad  | TI | 85 |
| 95655 | yayun | TD | 94 |
| 95712 | Johan | SD | 55 |
| 95777 | Johan | ED | 88 |
| 97885 | Yohan | MT | 87 |

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

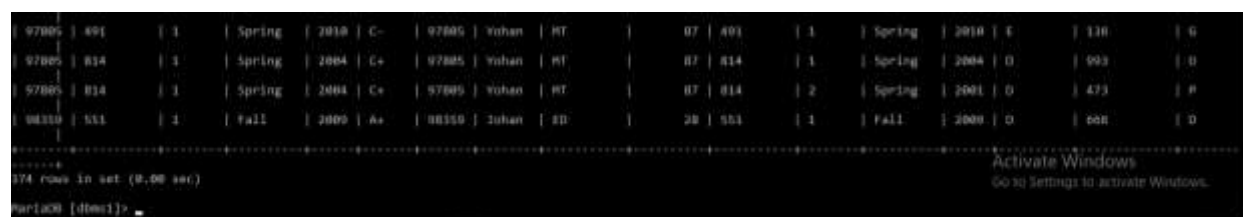


74 rows in set (0.09 sec)

|       |    |
|-------|----|
| Andi  | MT |
| Adri  | ED |
| Amad  | TI |
| yayun | TD |
| Johan | SD |
| Johan | ED |
| Yohan | MT |

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.



174 rows in set (0.08 sec)

|       |     |   |        |      |    |       |       |    |    |     |   |        |      |   |     |   |
|-------|-----|---|--------|------|----|-------|-------|----|----|-----|---|--------|------|---|-----|---|
| 97885 | 891 | 1 | Spring | 2010 | C+ | 97885 | Yohan | MT | 07 | 891 | 1 | Spring | 2010 | C | 138 | 0 |
| 97885 | 814 | 1 | Spring | 2004 | C+ | 97885 | Yohan | MT | 07 | 814 | 1 | Spring | 2004 | D | 993 | 0 |
| 97885 | 814 | 1 | Spring | 2004 | C+ | 97885 | Yohan | MT | 07 | 814 | 2 | Spring | 2004 | D | 473 | 0 |
| 98359 | 551 | 1 | Fall   | 2009 | As | 98359 | Johan | ED | 28 | 551 | 1 | Fall   | 2009 | D | 608 | 0 |

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

|       |    |   |        |     |   |     |    |
|-------|----|---|--------|-----|---|-----|----|
| Rudi  | ID | 1 | Fall   | 403 | B | 589 | GT |
| Rudi  | ID | 2 | Spring | 656 | D | 793 | GT |
| Rudi  | ID | 2 | Spring | 983 | D | 793 | GT |
| Yohan | MT | 1 | Spring | 136 | E | 273 | KL |
| Yohan | MT | 1 | Spring | 136 | E | 401 | ED |
| Yohan | MT | 1 | Spring | 983 | D | 814 | GO |
| Yohan | MT | 1 | Spring | 473 | D | 814 | GO |
| Yohan | ED | 1 | Fall   | 666 | D | 553 | ED |

174 rows in set (0.00 sec)

mysql>

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 99064 | Yohan | MT | 40 |
| 95954 | Yohan | MT | 23 |
| 95712 | Yohan | MT | 55 |
| 9577  | Yohan | MT | 10 |
| 97166 | Rudi  | ED | 30 |
| 97885 | Yohan | MT | 87 |
| 98359 | Yohan | ED | 28 |

8 rows in set (0.00 sec)

2. `SELECT * FROM student WHERE tot_cred > 30;`

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 04110 | Gede  | PT | 05 |
| 04062 | Adri  | ED | 38 |
| 05084 | Gema  | TI | 05 |
| 05055 | Yayan | TI | 05 |
| 05712 | Sonu  | SD | 52 |
| 05722 | Johan | ED | 88 |
| 07005 | Yohan | PI | 07 |

74 rows in set (0.00 sec)

```
Marlaud [dbms1]>
```

3. `SELECT `name`, `dept_name` FROM student WHERE tot_cred > 30;`  
Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

|       |    |
|-------|----|
| Gede  | PT |
| Adri  | ED |
| Gema  | TI |
| Yayan | TI |
| Sonu  | SD |
| Johan | ED |
| Yohan | PI |

74 rows in set (0.00 sec)

```
Marlaud [dbms1]>
```

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`  
Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

|       |     |   |        |      |    |       |       |    |    |     |   |        |      |   |     |   |
|-------|-----|---|--------|------|----|-------|-------|----|----|-----|---|--------|------|---|-----|---|
| 07005 | 273 | 1 | Spring | 2001 | A- | 07005 | Yohan | PI | 07 | 273 | 1 | Spring | 2001 | 1 | 130 | 0 |
| 07005 | 401 | 1 | Spring | 2010 | C- | 07005 | Yohan | PI | 07 | 401 | 1 | Spring | 2010 | 1 | 110 | 6 |
| 07005 | 814 | 1 | Spring | 2004 | C+ | 07005 | Yohan | PI | 07 | 814 | 1 | Spring | 2004 | 0 | 003 | 0 |
| 07005 | 814 | 1 | Spring | 2004 | C+ | 07005 | Yohan | PI | 07 | 814 | 2 | Spring | 2004 | 0 | 473 | P |
| 08332 | 551 | 1 | Fall   | 2009 | A+ | 08332 | Johan | ED | 28 | 551 | 1 | Fall   | 2009 | 0 | 000 | 0 |

174 rows in set (0.01 sec)

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`  
Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

|       |    |   |        |     |   |     |    |
|-------|----|---|--------|-----|---|-----|----|
| Budi  | ED | 1 | Spring | 138 | 8 | 293 | ED |
| Budi  | ED | 1 | Fall   | 463 | 8 | 386 | ED |
| Budi  | ED | 2 | Spring | 664 | 0 | 793 | ED |
| Budi  | ED | 2 | Spring | 903 | 0 | 793 | ED |
| Yohan | PI | 1 | Spring | 138 | 8 | 273 | ED |
| Yohan | PI | 1 | Spring | 138 | 8 | 401 | ED |
| Yohan | PI | 1 | Spring | 903 | 0 | 814 | ED |
| Yohan | PI | 1 | Spring | 473 | 0 | 814 | ED |
| Johan | ED | 1 | Fall   | 664 | 0 | 551 | ED |

174 rows in set (0.85 sec)

mariaDB [dbms1]>

Hasil untuk settings advisor = 10000, student = 10000, section = 30000000,takes = 30000000 seperti berikut. Dengan menjalankan query dibawah ini, didapatkan hasil antara lain :

1. Select \* From student

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 95655 | Yohan | PI | 88 |
| 95654 | Yohan | PI | 28 |
| 95712 | Johan | ED | 55 |
| 95777 | Johan | ED | 88 |
| 97165 | Budi  | ED | 38 |
| 97885 | Yohan | PI | 87 |
| 98359 | Johan | ED | 26 |

168 rows in set (0.86 sec)

2. SELECT \* FROM student WHERE tot\_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query kedua, dapat dilihat pada gambar dibawah ini.

|       |       |    |    |
|-------|-------|----|----|
| 95118 | Ande  | PI | 95 |
| 94682 | Andi  | ED | 38 |
| 95354 | Andi  | PI | 15 |
| 95655 | Yohan | PI | 88 |
| 95712 | Johan | ED | 55 |
| 95777 | Johan | ED | 88 |
| 97885 | Yohan | PI | 87 |

74 rows in set (0.89 sec)

mariaDB [dbms1]>

3. SELECT `name`, `dept\_name` FROM student WHERE tot\_cred > 30;

Waktu yang diperlukan dalam mengeksekusi query ketiga, dapat dilihat pada gambar dibawah ini.

|       |    |
|-------|----|
| Ande  | PI |
| Andi  | ED |
| Andi  | PI |
| Yohan | PI |
| Johan | ED |
| Johan | ED |
| Yohan | PI |

74 rows in set (0.89 sec)

mariaDB [dbms1]>

4. `SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id`

Waktu yang diperlukan dalam mengeksekusi query ke-empat, dapat dilihat pada gambar dibawah ini.

|       |     |   |        |      |    |       |       |    |    |     |   |        |      |   |     |   |
|-------|-----|---|--------|------|----|-------|-------|----|----|-----|---|--------|------|---|-----|---|
| 97805 | 491 | 1 | Spring | 2010 | C- | 97805 | Yohan | MT | 07 | 491 | 1 | Spring | 2010 | E | 136 | D |
| 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | MT | 07 | 814 | 1 | Spring | 2004 | D | 903 | D |
| 97805 | 814 | 1 | Spring | 2004 | C+ | 97805 | Yohan | MT | 07 | 814 | 2 | Spring | 2004 | D | 473 | P |
| 00350 | 551 | 1 | Fall   | 2000 | A+ | 00350 | Johan | ED | 20 | 551 | 1 | Fall   | 2000 | D | 008 | D |

374 rows in set (0.01 sec)

5. `SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

|       |    |   |        |     |   |     |    |
|-------|----|---|--------|-----|---|-----|----|
| Budi  | ID | 1 | Spring | 136 | E | 203 | ED |
| Budi  | ID | 1 | Fall   | 403 | E | 308 | GT |
| Budi  | ID | 2 | Spring | 606 | D | 793 | GT |
| Budi  | ID | 2 | Spring | 903 | D | 793 | GT |
| Yohan | MT | 1 | Spring | 136 | E | 273 | KL |
| Yohan | MT | 1 | Spring | 136 | E | 401 | ED |
| Yohan | MT | 1 | Spring | 903 | D | 814 | GO |
| Yohan | MT | 1 | Spring | 473 | D | 814 | GO |
| Johan | ED | 1 | Fall   | 006 | D | 551 | ED |

374 rows in set (0.01 sec)

Lalu pada file my.ini difolder xampp/mysql/bin, ganti isinya dengan format seperti tahap sebelumnya untuk melakukan index tuning. Lalu jalan kueri seperti sebelumnya :

1. `Select * From student`

Waktu yang diperlukan dalam mengeksekusi query pertama, dapat dilihat pada gambar dibawah ini.





5. `SELECT student.`name`,student.dept_name,takes.sec_id AS`

`pengambilan,takes.semester,section.room_number,section.building,course.course`

`_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN`

`section ON takes.course_id = section.course_id JOIN course ON`

`section.course_id = course.course_id`

Waktu yang diperlukan dalam mengeksekusi query kelima, dapat dilihat pada gambar dibawah ini.

|       |    |   |        |     |   |     |    |
|-------|----|---|--------|-----|---|-----|----|
| Sudi  | ID | 1 | Fall   | 463 | 0 | 586 | 01 |
| Sudi  | ID | 2 | Spring | 666 | 0 | 703 | 07 |
| Sudi  | ID | 2 | Spring | 993 | 0 | 793 | 07 |
| Yohan | MT | 1 | Spring | 136 | E | 273 | 01 |
| Yohan | MT | 1 | Spring | 136 | E | 401 | 02 |
| Yohan | MT | 1 | Spring | 983 | 0 | 814 | 00 |
| Yohan | MT | 1 | Spring | 873 | 0 | 814 | 00 |
| Johan | ID | 1 | Fall   | 666 | 0 | 551 | 00 |

274 rows in set (0.00 sec)

MariaDB [dewan]>

Berikut hasil yang didapatkan dikumpulkan dalam bentuk table :

|                                                                             | q1                        |                      | q2                   |                      | q3                   |                           | q4                        |                           | q5                        |                           |
|-----------------------------------------------------------------------------|---------------------------|----------------------|----------------------|----------------------|----------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| Data                                                                        | Waktu Sebelum Tuning (ms) | Waktu Sesudah Tuning | Waktu Sebelum Tuning | Waktu Sesudah Tuning | Waktu Sebelum Tuning | Waktu Sesudah Tuning (ms) | Waktu Sebelum Tuning (ms) | Waktu Sesudah Tuning (ms) | Waktu Sebelum Tuning (ms) | Waktu Sesudah Tuning (ms) |
| advisor = 100,<br>student = 100,<br>section = 200,takes = 200               | 0 second                  | 0.02 S               | 0 second             | 0 second             | 0 second             | 0.0 second                | 0.01 second               | 0.0 second                | 0.05 second               | 0.01 second               |
| advisor = 200,<br>student = 200,<br>section = 400,takes                     | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0 second                  | 0.01 second               | 0 second                  | 0 second                  |
| advisor = 500,<br>student = 500,<br>section = 1000,takes                    | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0.01 second               | 0.1 second                | 0 second                  | 0.01 second               |
| advisor = 700,<br>student = 700,<br>section = 20000,takes                   | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0.01 second               | 0 second                  | 0 second                  | 0 second                  |
| advisor = 1000,<br>student = 1000,<br>section = 100000,takes =              | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0 second                  | 0 second                  | 0 second                  | 0 second                  |
| advisor = 1800,<br>student = 1800,<br>section = 180000,takes =              | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0 second                  | 0.01 second               | 0 second                  | 0.01 second               |
| advisor = 10000,<br>student = 10000,<br>section = 30000000,takes = 30000000 | 0 second                  | 0 second             | 0 second             | 0 second             | 0 second             | 0 second                  | 0.01 second               | 0 second                  | 0 second                  | 0 second                  |

## **Bab III Kesimpulan**

### **III.1. Kesimpulan**

Dari hasil yang didapat, telah dilakukan tuning dengan sistem indexing pada database dan mendapatkan waktu eksekusi query yang lebih singkat. Hal ini sesuai dengan fungsi performance tuning dengan metode indexing dimana berperan untuk meningkatkan performa DBMS yang digunakan.